Chapter 5. Outbound/Inbound and Outbound Structured Fields

Introduction

This chapter describes the outbound/inbound and outbound structured fields that can be included in the 3270 data stream. To provide additional controls and transmit various data types other than characters, it is necessary to use structured fields. The structured field syntax permits variable length data and controls to be encoded in such a way as to facilitate processing a sequence of fields into component fields without having to examine every byte.

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

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

length 1	ID	information	length 2	ID	information

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

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

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

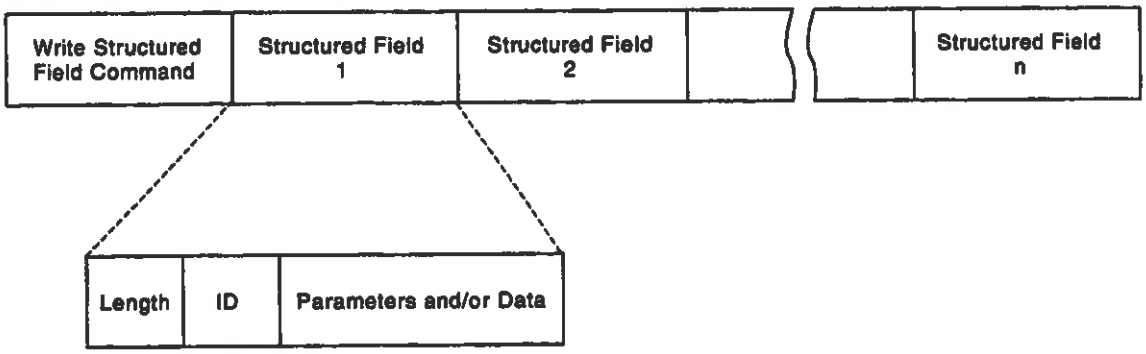


Figure 5-1. Structured-Field Format

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

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

Except for the use of a Length parameter value of zero, a structured field with a one-byte type parameter will be rejected if the Length field value is less than three. For a structured field with a two-byte type field, the minimum Length field value is four. This means that a structured field requires at least an ID. Thus, sending only the Length field (i.e. 0000), as a zero length structured field is invalid.

The outbound structured fields are described under "Outbound Structured Fields" on page 5-11. The fields and their ID codes are shown in alphabetical order in the following list. Major operations within the field are listed by their operation code.

ID	Name	ID	Name
0E	Activate Partition	01*nn*	Read Partition
0F85	Begin/End of File	*nn*=	02 Query
0C	Create Partition	*nn*=	03 Query List
0D	Destroy Partition	*nn*=	6E Read Modified All
03	Erase/Reset	*nn*=	F2 Read Buffer
0F05	Load Color Table	*nn*=	F6 Read Modified
0F24	Load Format Storage	1030	Request Recovery Data
0F07	Load Line Type	00	Reset Partition
06	Load Programmed Symbols	1033	Restart
0F0A	Modify Partition	41	SCS Data
0F71	Outbound Text Header	0F04	Select Color Table
40	Outbound 3270DS	4A	Select Format Group
4B	Present Absolute Format	1032	Set Checkpoint Interval
4C	Present Relative Format	0F01	Set MSR Control
0F08	Set Partition Characteristics	0F84	Set Printer Characteristics
09	Set Reply Mode	0B	Set Window Origin
0FC1	Type 1 Text Outbound		

The outbound/inbound structured fields are described under "Outbound/Inbound Structured Fields" on page 5-72. The fields and their ID codes are shown in alphabetical order in the following list:

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

Structured Field Grouping

Structured fields cannot span transmission. An error occurs if a transmission ends before a structured field's length count is satisfied or if an entire control sequence does not appear in the same structured field. However, you can group related structured fields into a single logical entity. When structured fields are grouped as described in this section, the non-spanning rules apply on a group basis rather than on a structured field basis.

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

B'00' Continue
B'01' End
B'10' Begin
B'11' Only.

In the first structured field of a group, the Group parameter is set to Begin. The group parameter is set to Continue in the second through the next-to-last structured fields of the group. The group parameter is set to End in the last structured field of the group. When the structured field is not a member of a group of related structured fields, the group parameter is set to Only; the non-spanning rules apply in this case.

In addition to the explicit termination of a structured field group by a group parameter set to End, the following terminates (without error) all groups in process to or from the device:

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

Spanning

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

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

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

- The length count of the last structured field in a transmission must be satisfied at the end of the transmission. That is, a structured field cannot span transmissions (except if the Data Chaining structured field is used in conjunction with spanning, limited to the non-SNA environment).
- In SNA on outbound (to the device) more than one transmission can be sent without sending a CD until the last transmission of the group.

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

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

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

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

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

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

- Receiving a structured field with the Group parameter = Continue or End when the group is not in existence.
- Receiving a structured field with the Group parameter = Begin for a partition that already has a group, with the same structured type, in process.
- Receiving an individual (non-grouped) structured field for a partition which has a group, with the same structured field type, in process.

- Receiving a transmission starting with any 3270 command except WSF or EW/EWA (with WCC = Reset).
- A structured field length count is not satisfied at the end of a transmission.

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

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

Data Chaining (Non-SNA)

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

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

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

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

The Data Chain structured field in the first transmission has the Group parameter = Begin. In the next through the next-to-last transmission, the Group parameter is set to = Continue. In the last transmission, the Group parameter = End. In the first transmission, a structured field must start immediately following the Data Chain structured field. However, in subsequent transmissions, this is not required. The data following can be a continuation of a structured field started in a previous transmission. That is, the length count of a structured field can span transmissions. If the length count of a structured field is not satisfied at the end of a transmission (excluding the last transmission of a data chain), the remainder of

the data to satisfy the count starts immediately *after* the Data Chain structured field in the next transmission. The length count of a structured field could in some cases carry across a number of transmissions.

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

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

- Receiving a Data Chain structured field with Group parameter = Continue or End without having previously started a data chain with a Data Chain structured field with the Group parameter = Begin
- Receiving a Data Chain structured field with the Group parameter = Begin while a data chain already exists
- Receiving a transmission that starts with any 3270 command except a Copy (BSC only), a WSF (followed by a valid Data Chain structured field) or an EW/EWA (with the WCC = Reset).

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

- Sending a Data Chain structured field with the Group parameter = Continue or End without having first begun the data chain with a Data Chain structured field with the Group parameter = Begin
- Sending a Data Chain structured field with the Group parameter = Begin while a data chain already exists.

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

- Receiving an outbound Data Chain structured field with INCTRL = Inbound Data Chaining Disabled
- Receiving an EW/EWA command with WCC = Reset
- POR
- Clearing a local function (for instance, pressing the Clear key).

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

Structured Field Self-Defining Parameters

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

Format of Self-Defining Parameters

All self-defining parameters have the following format:

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

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

Additional Content Description

Parameter length is a 1-byte binary value in the range 2 to 127. Values less than 2 are invalid; values greater than 127 are reserved (the high-order bit of the length byte must be zero). The length value includes 1 byte for Parameter Length and 1 byte for the Parameter Identifier Value. There is no default value.

Parameter identifier value is a 1-byte binary value that characterizes the information content of this self-defining parameter. It can have any value listed in the outlined value list. There is no default value.

Parameters are variable length fields that contain data needed to perform the function specified by the Parameter Identifier Value. This field can be from zero to 125 bytes long.

Outbound Structured Fields

The outbound structured fields (application to the device) are defined in the following pages. Structured field type codes not shown are rejected. Unless specifically stated to the contrary, any bits or fields classified as reserved are checked for zero value. Nonzero values are rejected (see Appendix A, "SNA Sense Codes").

Activate Partition

The Activate Partition structured field activates a partition.

Function

The Activate Partition structured field functions as follows:

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

Format

The following table shows the format of the Activate Partition structured field:

Bytes	Content	Content	Description
0-1	L	X'0004'	Length of this structured field
2	SFID	X'0E'	Identifies this structured field as Activate Partition
3	PID	Partition identifier (X'00' through X'7E')	

Begin/End of File

This section describes the Begin/End of File outbound structured field.

Function

The Begin/End of File structured field indicates to the device that a new file is beginning or that the current file is completed.

Format

The following table shows the format of the Begin/End of File structured field: scale='.9' arrange='1 2 3 4 / 5 6 7 8' mindepth=p5 frame=box hdframe=rules split=yes. frame=box hdframe=rules split=yes.

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

Create Partition

This section describes the Create Partition outbound structured field.

Function

The Create Partition structured field creates a new partition.

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

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

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

The conditions shown below occur whether the display is in the implicit or explicit partitioned state. Parameter references are to the Create Partition Structured Field Base description which follows the list below:

- The viewport of the new partition is assigned to origin (RV, CV) relative to the usable area, width WV and height HV. If any of the following conditions exist, the structured field is rejected:
  - WV=0 and modify partition is not supported.
  - HV=0 and modify partition is not supported.
  - RV+HV > useable area height if the Unit of Measure (UOM)=X'0' or X'1' or RV+(HV+PH) > useable area height (points) if UOM=X'2'
  - CV+WV > useable area width if the UOM=X'0' or X'1' or CV+(WV×PW) > useable area width (points) if UOM=X'2'.
  - The viewport overlaps another viewport.
  - The viewport cannot be contained within the usable area.
- The current buffer address for the new partition is set to zero.
- The current cursor position for the new partition is set to the window origin.
- The nonloadable character set is established from the list returned in the character sets Query Reply according to the value of BASE.

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

Format

The following table shows the format of the Create Partition structured field:

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

Byte	Bit	Content	Content Description
28-29		PH	The number of points in the vertical direction in a character cell in this presentation space

Working backwards from byte 29, the previously defined parameters can be progressively omitted by specifying the appropriate length of the structured field.

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

The following table shows the defaults for omitted parameters:

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

Additional Content Description

The content of the Create Partition structured field is further described as follows:

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

For this UOM value:	Parameters are interpreted as follows:
B'0000'	In the row, column coordinate system
B'0001'	In the all points addressable system
B'0010'	In the row, column coordinate system except for fields RV and CV, which are interpreted in the all points addressable system.

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

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

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

A-MODE If A-MODE is B'0010', a default text environment is established (Adjust off, no Tabs, text width = W, and left margin = 1).

RS If RS is X'FFFF', vertical scrolling is device dependent. The device determines the amount of data movement for explicit requests from the operator and for automatic scrolling.

FLAGS Flags may be specified as follows:

PROT If the PROT flag is specified as B'1', the newly created partition is marked as protected.

COPY The COPY flag determines whether the viewport or presentation space are locally copied. If the copy flag is set to B'0', the contents of the character buffers and attribute buffers (if any) that represent the viewport of the active partition are utilized. If the COPY flag is B'1', the contents of the character buffers and attribute buffers (if any) that represent the presentation space of the active partition are utilized. See Chapter 8, "Printer Operations" for details of local copy operation.

BASE The BASE parameter is used to select the appropriate base character set for a partition by indexing down the list of base character set descriptors in the order returned in the Character Sets Query Reply.

Notes:

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

Define Presentation Space Types Self-Defining Parameter

The Define Presentation Space Types self-defining parameter defines the presentation spaces for this partition.

The presentation space types are always ordered Alphanumerics, Image, and Graphics with Alphanumerics as the topmost layer.

Notes:

1. If an Image-only presentation space is created, the UOM must be addressable points.
2. If this self-defining parameter is omitted, an Alphanumeric presentation space will be created (the default is X'00').

The following table shows the format of the Define Presentation Space Types self-defining parameter:

Byte	Content	Content Description	
0	L	Length	
1	**SDPID**	**X'05'**	Identifies this self-defining parameter as Define Presentation Space Types
2	**FLAGS**	Reserved	
3-n	**PTYPES**	PS types defined	
		X'00'	Alphanumerics
		X'01'	Reserved
		X'02'	Image
		X'03'	Graphics

Destroy Partition

This section describes the Destroy Partition outbound structured field.

Function

The Destroy Partition structured field destroys a named partition. If the named partition is the only existing partition, the Destroy Partition structured field creates default implicit partition zero. *Destroy* means the act of doing the following:

1. Removing the PID from further accessibility
2. Prohibiting any further display from the character buffer that had been assigned to the PID until that buffer space is erased or overwritten
3. Clearing or overwriting the displayed viewport associated with the PID.

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

If the named partition is the only existing partition, Destroy Partition does the following:

1. Destroys the named partition
2. Resets the display to implicit partition state and creates the default implicit partition zero
3. Resets any input-inhibit conditions
4. Resets INOP to Read Modified and INPID to zero.

If the named partition is not the only existing partition, Destroy Partition does the following:

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

      If it has a zero extent window, the cursor is displayed in the right hand end of the operator information area.

   b. Any saved PWAIT or SYSTEM LOCK conditions for that partition are raised, and the appropriate indicators are displayed.
   c. If the specified PID = INPID, then INPID is reset to zero, and INOP is reset to Read Modified.
2. Destroys the named partition.

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

Format

The following table shows the format of the Destroy Partition structured field:

Byte	Content	Content Description	
0-1	L	X'0004'	Length of this structured field
2	SFID	X'0D'	Identifies this structured field as Destroy Partition
3	PID	The partition identifier X'00' through X'7E'	

Erase/Reset

This section describes the Erase/Reset outbound structured field.

Function

The Erase/Reset structured field resets the device to implicit partition state destroying all existing (implicit or explicit) partitions. The function creates an implicit partition zero with default partition characteristics and of default size if IPZ = B'0' or of alternate size if IPZ = B'1'.

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

Format

The following table shows the format of the Erase/Reset structured field:

Byte	Bit	Content	Content Description
0-1		L	X'0004' Length of this structured field
2		SFID	X'03' Identifies this structured field as Erase/Reset
3		FLAGS	
	0	IPZ	Implicit partition size: B'0' Default size B'1' Alternate size
	1-7	RES	Reserved

Load Color Table

This section describes the Load Color Table outbound structured field.

Function

The Load Color Table structured field defines a loadable color table.

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

Format

The following table shows the format of the Load Color Table structured field:

Byte	Bit	Content	Content Description
0 - 1		**L**	Length of this structured field
2 - 3		**SFID**	**X'0F05'** Identifies this structured field as Load Color Table
4 - N		**COMMAND**	Command

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

Load Format Storage

This section describes the Load Format Storage outbound structured field.

Function

The Load Format Storage structured field provides the means of distributing 3270 data stream display formats to the Format Storage Auxiliary Device and requesting reports of Format Storage Auxiliary Device storage utilization. The formats are stored in the control unit and are accessed by the Select Format Group, Present Absolute Format, and Present Relative Format structured fields.

Format

The following table shows the format of the Load Format Storage structured field:

Byte	Bit	Content	Content Description
0-1		**L**	Length of this structured field
2-3		**SFID**	**X'0F24'** Identifies this structured field as Load Format Storage
4		**FLAGS**	
	0-1	RES	Reserved, must be set to B'11'.
	2	FSEL	Format Selection: **B'0'** Format is not user selectable. **B'1'** Format is user selectable.
	3	SS	Screen Size for user selected formats: **B'0'** Default screen size. **B'1'** Alternate Screen Size.
	4-7	RES	Reserved, must be set to zeros.
5		**FLAGS**	Reserved, must be set to zeros.
6		**OPRND**	Operand: **X'01'** Add **X'02'** Delete Format **X'03'** Delete Group **X'04'** Reset All **X'05'** Request Summary Status **X'06'** Request Group Status
7-14		**LCLNME**	Local name for user selectable formats
15-30		**FGNME**	Format Group name
31-46		**FNME**	Format name
47-n		**FMTDATA**	The format to be stored

With the exception of the FMTDATA field, all the parameters shown must be present. If the OPRND parameter = ADD, then the FMTDATA must also be present. If the OPRND parameter is any other valid value, the FMTDATA field is not present.

The names provided by the LCLNME, FGNME, and FNME parameters are in EBCDIC.

Additional Content Description

The content of the Load Format Storage structured field is further described as follows:

L These bytes specify the length of any structured field. The length must be explicitly specified and therefore the bytes are not allowed to be sent with a value of zero.

FSEL This flag indicates whether or not the user can select the format in addition to host application selection. Host selection is accomplished by the Select Format Group, Present Absolute Format, or Present Relative Format structured fields. An FSEL value of B'1' indicates that both the user and host application selection of the format is allowed.

SS This flag indicates whether the format applies to the default or to the alternate screen size. The screen sizes referred to are those associated with the device returning the Format Storage Auxiliary Device Query Reply. This flag applies only when the format is selected by the user.

OPRND Indicates the operation the structured field performs. The operands are:

ADD The structured field adds a format to format storage if a format of the same name does not already exist in format storage. If a format of the same name already exists, the contents of that format are replaced by the contents of the FMTDATA field.

If the group directory does not exist, it is created.

If a local name for user selected formats is specified, it is catalogued in the local names directory.

Available status is returned if the Add operation is successful. Exception status is returned if there is insufficient storage to add the format or if group name or format name is omitted.

DELETE FORMAT

This deletes the format named in the FNME field.

If a local name exists for the deleted format, it is removed from the local names directory.

Available status is returned if the Delete operation is successful or if the specified format is not found. Exception status is returned if the group name or format name is omitted.

DELETE FORMAT GROUP

This deletes the format group named in the FGNME field. The associated group directory is deleted and affected local names are removed from the local names directory.

If no group name is specified, exception status is returned. If the specified group name is not found, available status is returned.

RESET ALL

This deletes all formats in all format groups and clears the local names directory. Available status is then returned.

REQUEST SUMMARY STATUS

Indicates a request for the device to report the following status by the Format Status self-defining parameter of the Exception/Status structured field:

- Number of groups assigned
- Total number of formats loaded
- Total number of local formats used
- The amount of format storage spaces available.

REQUEST GROUP STATUS

Requests the device to report the number of formats currently loaded in the group specified by the FGNME parameter. The device reports the requested information by the Group Status self-defining parameter of the Exception/Status structured field.

If the request is made for a non-existent group name, then the number of formats is set to zero.

LCLNME Provides a local name for the format that the user can use for selection. LCLNME (and the later FGNME and FNME) are in the form of alphanumeric characters. For example, the name 'BA-----' would appear as X'C2C1-----'. All the bytes must be in the range of X'40' through X'FE'. The LCLNME parameter is applicable only when the FSEL flag indicates user selection of formats is allowed (FSEL = B'1'). If the FSEL flag indicates user selection of formats is not allowed (FSEL = B'0'), LCLNME must be set to zeros.

If a local name for user names is specified, it is catalogued in the local names directory.

FGNME Provides the Format Group name. When OPRND = ADD or DELETE, it indicates the Format Group to which the format is to be added or deleted. When OPRND = Delete Group, it indicates the Format Group to be deleted. This parameter is ignored if OPRND = Reset All.

If the group name is omitted, exception status is returned.

FNME Provides the name of the format. When OPRND = ADD, it provides the name assigned to the format being loaded. When OPRND = Delete Format, it indicates the name of the format to be deleted. This parameter is ignored if OPRND = Delete Group or Reset.

If the format name is omitted for Add or Delete, exception status is returned.

FMTDATA This is the actual format being added to format storage. It is present only when the OPRND parameter = ADD. It is an error condition if present when OPRND is other than ADD.

The data comprising the format is exactly the same as would be sent to a 3270 data stream display if the format were to be displayed immediately rather than stored for future use. A Load Format Storage structured field can contain only *one* format.

Load Line Type

This section describes the Load Line Type outbound structured field.

Function

The Load Line Type structured field specifies the controlling environment defined line types.

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

Format

The following table shows the format of the Load Line Type structured field:

Byte	Bit	Content	Content Description
0-1		L	Length of this structured field
2-3		SFID	X'0F07' Identifies this structured field as Load Line Type
4-n		COMMAND	Command

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

Load Programmed Symbols (Load PS)

This section describes the Load Programmed Symbols outbound structured field.

Function

The Load Programmed Symbols structured field loads symbol and character definition information into the device.

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

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

Format

The following table shows the format of the Load Programmed Symbols structured field:

Byte	Bit	Content	Content Description
0-1		L	Length of this structured field
2		SFID	X'06' Identifies this structured field as Load Programmed Symbols (Load PS)
3		FLAGS	
	0	Basic/Ext	Basic or extended form: B'0' Basic form: only bytes 0 through 6 present B'1' Extended form: length determined by byte 7
	1	CLEAR	Clear loadable character set: B'0' Do not clear loadable character set. B'1' Clear all character slots not loaded.
	2	SKIP	Skip suppress (Note 1): B'0' Suppression not required B'1' Suppression required
	3-7	TYPE	Data format type: B'00000' Reserved B'00001' (Type 1) 18-byte form; the first 2 bytes contain a 16-bit vertical slice, the following 16 bytes contain 8-bit horizontal slices. For a 9 x 12 character matrix the last 4 bytes contain binary zero.

Byte	Bit	Content	Content Description
			B'00010' (Type 2) Type 1 compressed **B'00011' (Type 3)** Row loading (from top to bottom) **B'00100' (Type 4)** Type 3 compressed **B'00101'(Type 5)** Column loading (from left to right) **B'00110' (Type 6)** Type 5 compressed **B'01000' (Type 8)** Vector **Others** Reserved
4		**LCID**	Local character set ID: Use X'40' to X'EF'. Use X'FF' to indicate that RWS associated with this LCID is free (not assigned). Other values are reserved.
5		**CHAR**	Beginning code point X'41' through X'FE'
6		**RWS**	Loadable Character Set RWS Number

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

Byte	Bit	Content	Content Description
7		**P Length**	Length of parameters for extended form, including the length parameter. The parameters defined below can be progressively included by specifying the appropriate length. Omitted parameters are assumed to have the defined default indicated by the zero value for that parameter.
8		**FLAGS**	
	0	APA	All points addressable (APA): **B'0'** APA **B'1'** Not APA
	1	CB	LCID compare bit: **B'0'** LCID compare **B'1'** No LCID compare
	2	OB	Operator selectable by PS key: **B'0'** Operator selectable **B'1'** Not operator selectable
	3	MULTID	Multiple LCID: **B'0'** Release any other symbol set with the same LCID. **B'1'** Release any other symbol set with the same LCID unless it was loaded with MULTID=B'1'.

Byte	Bit	Content	Content Description
	4	USET	Use Symbol Envelope Table. **B'0'** Do not use S.E.T. information. **B'1'** Use S.E.T. information.
	5-6	RES	Reserved
	7	SDPP	Self-Defining Parameters Present: **B'0'** No self-defining parameters present **B'1'** Self-defining parameters follow
9		**LW**	Number of X-units in character cell (width of character matrixes)
10		**LH**	Number of Y-units in character cell (depth of character matrixes)
11		**SUBSN**	Subsection ID: **X'00'** One-byte codes **X'41'- X'FE'** Subsection ID for 2-byte coded data **Other** Reserved
12		**COLOR**	Color planes: **X'00'** All planes **X'01'** Blue plane **X'02'** Red plane **X'04'** Green plane **X'nn'** Color attribute value
13		**ST. SUBS**	Starting Subsection Identifier: **X'00' (default)** Starting Subsection defined by QUERY REPLY—CHARACTER SETS. **X'41'- X'FE'** Starting Subsection Identifier for 2-byte coded character set specified by LCID in this Load PS. **Others** Reserved
14		**ECHAR**	Ending code point
15		**NW**	Number of width pairs
16		**NH**	Number of height pairs
17		**RES**	Reserved - must be zero

Notes:

1. Values for parameters other than those shown above are rejected; invalid parameter values are also rejected. (See Table A-1 on page A-1.)
2. The lost space resulting from setting of the skip suppress to B'1' is made up following the last display/print line; that is, the last display/print line moves up from the bottom of the physical presentation space. For a printer, setting of skip suppress to B'1' must not result in loss of forms sync following a form feed (FF) or forms overflow.

The PS data follows in the format described in bits 3-7 of byte 3.

Byte	Bit	Content	Content Description
M to N		DATA	Character matrices

Additional Content Description

The content of the Load Programmed Symbols structured field is further described as follows:

BASIC/EXT The extended-form bit in byte 3 indicates whether the structured field is of the extended form. The extended form contains additional information associated with copy operations, character matrix size, and color. If the device does not support the extended form and this bit is set to 1, the data stream is rejected with a sense code of X'1005' or X'1001'.

SKIP If skip suppression is on, any row in the usable area containing characters from this character set has the skip suppressed that would normally follow that row. The characters on the following row are then vertically adjacent to characters on the current row. If the skip suppression flag is not set, no skip suppression is required for characters from this character set

CLEAR If the Clear flag is set on, all slots in the specified RWS are cleared. The character definitions in the data portion of the structured field, interpreted according to the data format type (byte 3), are then loaded into contiguously addressable slots in the PS RWS, starting at the position defined by Char.

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

TYPE If the data format type specifies *compressed* data (type 2, 4, or 6), the data is first decompressed. (See "Compression Function" on page 5-34.)

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

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

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

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

- The data is exhausted. In this case the last complete cell definition in the data is loaded If there are any excess bits, they are ignored and -RSP (with a sense code of X'1001' or X'1005') is returned.
- The slot corresponding to X'FE' is loaded. Excess data is ignored and -RSP (with a sense code of X'1001' or X'1005') is returned.

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

The PS RWS number specified in this structured field is released from any previously assigned LCID. Any subsequent reference to this released LCID causes an error condition. The LCID and PS RWS numbers specified in this structured field are associated for any subsequent data stream processing.

An LCID of X'FF' indicates that this PS RWS is free (not assigned). The operator cannot select a PS RWS with an LCID = X'FF'.

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

RWS Byte 6 indicates the physical RWS to be loaded. There is a fixed relationship between the physical RWS number and the attribute selection keys defined for PS. The value in CHAR indicates the first slot to be loaded and must be in the range X'41' through X'FE'. If CHAR is outside this range, the data stream is rejected (invalid parameter, sense code X'1005' or X'1001').

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

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

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

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

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

When set to 1, the LCID of this character set is not compared with LCIDs of character sets in the printer. The copy operation is performed using the base character set of the printer.

OB When set to B'1', it signifies that this character set is intended for output only. Thus, the PS key normally associated with the RWS containing this character set cannot be enabled by the terminal operator while the key is connected to this LCID. Selection of individual character sets can, therefore, be disabled, even though the Set Reply Mode structured field allows character set selection. When the Set Reply Mode structured field disables character set selection, selection is disabled for all character sets independent of the setting of the OB bit for each character set.

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

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

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

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

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

COLOR A character set with triple-plane capability has three color planes into which characters can be loaded. For any code point (X'41' to X'FE') within the character set, each plane can be loaded independently. That is, a different character definition can be loaded into each of the primary planes in the character set for that code point.

For a triple-plane character set, if the Color field is B'001', B'010', or B'100', the character set data is loaded only into the specified character slots in that plane. Other color values are reserved and rejected.

For a triple-plane character set, if the Color field bits are omitted or if the Color field is B'000', then the character set data is loaded into the specified character slots in all three planes.

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

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

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

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

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

Symbol Envelope Table Self-Defining Parameter

The Symbol Envelope Table self-defining parameter provides the information pertinent to a single symbol. Therefore, there can be as many of these self-defining parameters as necessary. The first of these is the information for the symbol identified by the Char parameter. Every symbol of the set has the same numbers of indentation pairs. If this command creates a Symbol Envelope Table for this symbol store, then any symbols not assigned explicit values assume all zero values. Subsequent Load PS commands only alter the pair values (subject to the Clear parameter) for code points for which values are explicitly provided.

The following table shows the format of the Symbol Envelope Table self-defining parameter:

Byte	Content	Content Description
0	**L**	Length of parameter
1	**SDPID**	**X'01'** Identifies this self-defining parameter as Symbol Envelope Table
2 to 1 + 2NW	**WPAIRS**	Pairs of width indentation values.
2+2NW to 2NW +2NH+1	**HPAIRS**	Pairs of height indentation values.

The content of the Symbol Envelope Table self-defining parameter is further described as follows:

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

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

Terminator Self-Defining Parameter

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

The following table shows the format of the Terminator self-defining parameter:

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

Compression Function

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

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

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

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

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

Character Cell Division

The character cell for a display or printer character position is divided into slices as shown in Figure 5-2 on page 5-35 and in Figure 5-3 on page 5-36. A slice corresponds to a byte, and the bits to the pels of the character cell.

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

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

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

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

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

Compression Process

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

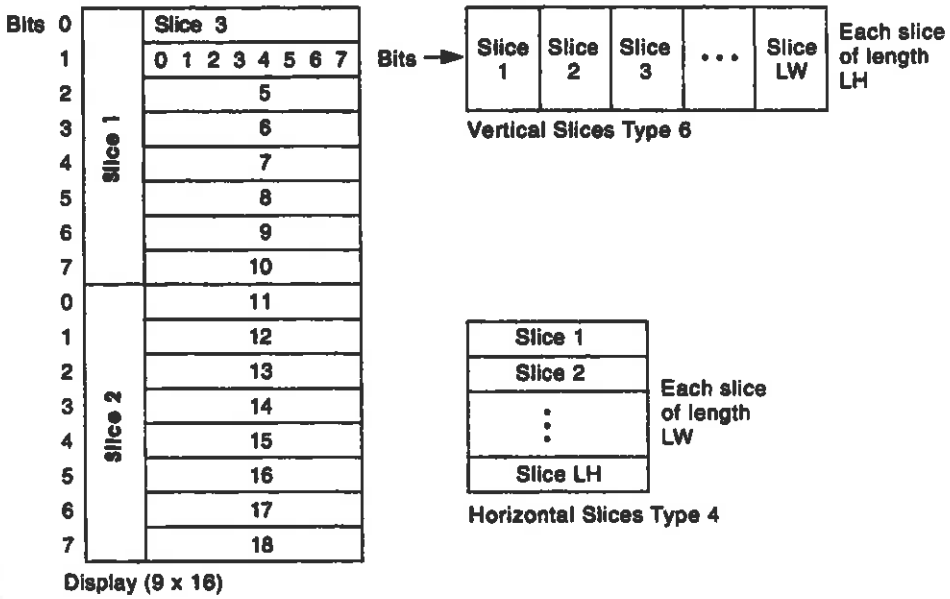


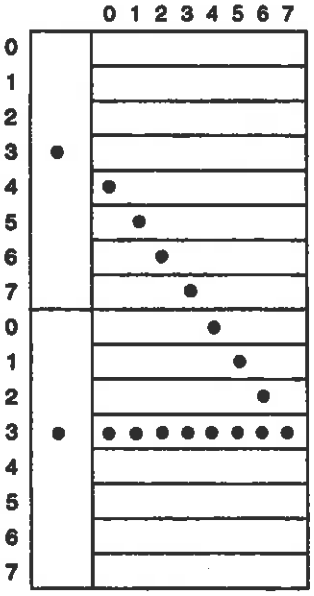
Figure 5-2. Vertical and Horizontal Slicing of a Character Cell

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

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

H Header bits
SD Symbol definition bits
T Terminator bits.

"Comparison Rules and Header Bits" describes the comparison rules and header bits, the creation of the compressed bit string, the terminator bits, and provides examples of compressing the symbol definition bit strings for the symbol in Figure 5-3.



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

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

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

Comparison Rules and Header Bits

The four comparison rules that follow are used in creating a compressed symbol definition bit string from a Type 1 (display) or Type 5 (printer) uncompressed symbol definition bit string. Encoding the results of the comparisons is discussed in "Creating the Compressed Bit String" on page 5-37.

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

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

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

  Each element is compared with the corresponding element in the previous slice; for example, the first element of slice 2 is compared with the first element of slice 1, the second element of slice 2 with the second element of slice 1, the first element of slice 3 with the first element of slice 2, and so on.

Since slice 1 has no previous slice, each element of slice 1 is compared with a 0 element.

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

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

- Comparison Rule 4 (Header bits = B'1110')

  The symbol definition consisted of 0 bits only and no comparison is required. It is a blank symbol.

If all of the header bits contain anything other than the bit combinations listed here, the data stream is rejected.

Creating the Compressed Bit String

The element comparisons are encoded by taking the elements four at a time (two slices — the slice pair referred to in the following discussions) and comparing them with their corresponding elements in a reference slice pair that is created (following the comparison rules). Because the elements are compared four at a time, it is convenient to regard the 18 slices of the Type 1 data string or the 10 slices of a Type 5 data string symbol definition as being made up of 9 or 5 slice pairs. The elements of the slice pairs are compressed as follows:

1. Compare the first Type 1 or 5 slice pair with the reference slice pair. If the two slice pairs are identical, put a 0 bit in the symbol definition bit string and repeat step 1 for the next slice pair. If the two slice pairs are not identical, put a 1 bit in the symbol definition bit string and continue with step 2.
2. Compare, in turn, each element in the Type 1 or 5 slice pair with the corresponding element in the reference slice pair. For each element that matches (that is, if the elements being compared are the same), put a 0 bit in the symbol definition bit string. For each element that does not match (that is, if the elements being compared are not the same), put a 1 bit in the symbol definition bit string followed by a copy of the four bits of the non-matching element from the Type 1 or 5 slice pair.
3. Repeat steps 1 and 2 through the ninth slice pair of a Type 1 string or the fifth slice pair of a Type 5 string.

Terminator Bits

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

Examples of the Compression Algorithm in Use

The following three examples show how a Type 1 data string for a particular symbol is compressed into a Type 2 data string. In these examples, the symbol whose Type 1 data string is being compressed is the one shown in Figure 5-3 on page 5-36.

The data string for that symbol presented as nine slice pairs is as follows:

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

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

Using Comparison Rule 1

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

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

With comparison rule 1, the header is B'0' and the symbol definition bit string is created by comparing each Type 1 slice pair with an all-zeros reference slice pair as shown in Figure 5-4 on page 5-38. The resulting bit string, including the header, is as follows:

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

The original Type 1 bit string of 144 bits is compressed into 74 bits.

Using Comparison Rule 2

Slice-Pairs Being Compared according to Rule 2	
Reference Slice-Pair	Type 1 Slice-Pair
X'0010'	X'1010'
X'1000'	X'0000'
X'0000'	X'0000'
X'0080'	X'8040'
X'4020'	X'2010'
X'1008'	X'0804'
X'0402'	X'02FF'
X'FF00'	X'0000'
X'0000'	X'0000'

Step 1: Compare slice-pairs. For a match, generate B'0'. For a mismatch, generate B'1' and do step 2.				
	Step 2: Compare elements. For a match, generate B'0'. For a mismatch, generate B'1' followed by a copy of the bits in the nonmatching Type 1 element.			
	Element 1	Element 2	Element 3	Element 4
1	1 0001	0	0	0
1	1 0000	0	0	
0				
1	1 1000	0	1 0100	0
1	1 0010	0	1 0001	0
1	1 0000	1 1000	0	1 0100
1	0	1 0010	1 1111	1 1111
1	1 0000	1 0000	0	0
0				

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

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

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

The original Type 1 bit string of 144 bits is compressed to 95 bits.

Using Comparison Rule 3

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

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

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

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

The original Type 1 bit string of 144 bits has been compressed to 112 bits.

Modify Partition

This section describes the Modify Partition outbound structured field.

Function

The Modify Partition structured field varies the partition parameters such as protection, viewport position, or window origin.

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

- The named partition does not exist.
- A reserved bit in a flag byte is not zero.
- The viewport overlaps another viewport.
- The viewport cannot be contained within the usable area.
- The window is not completely contained in the presentation space.
- Bytes 8-11 are not X'00000000'.

Working backward from byte 31 through byte 6, parameters can be omitted progressively by specifying the appropriate length. If a parameter is omitted in this way, its value is not altered. The specified values replace those currently in effect.

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

If the window origin changes and the cursor is outside the new window, the cursor moves by the minimum number of rows or columns so that it is within the new window.

If the size of the window is varied to zero and this partition is Active, the next partition (if any) with a nonzero window is activated. If every partition has a zero extent window, this partition remains active but the operator cannot type data into it, and the screen cursor is placed at the right-hand end of the indicator row.

If, at the end of a transmission, the display is in send or contention state and the active partition has a zero extent window, an error situation exists.

If this is the only partition with a window of nonzero extent, then it is Activated.

Some reserved fields in Modify Partition correspond to parameters in Create Partition. These partition parameters (for example, A-MODE) cannot be modified.

Format

The following table shows the format of the Modify Partition structured field:

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

Each parameter is described under "Create Partition" on page 5-13.

Outbound Text Header

This section describes the Outbound Text Header structured field.

Function

The Outbound Text Header structured field establishes initial conditions for a text partition.

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

- The partition A-MODE is not X'2'.
- The named partition does not exist.
- The OP-Type does not specify Write or EW.
- Reserved flag bits are not B'0'.
- LHDR < 2.
- The HDR contains any graphics or unsupported controls.
- The HDR contains inconsistent formatting controls (for example, LM > width).
- A parameter error occurs in any of the HDR controls.
- CC > W-1.

Format

The following table shows the format of the Outbound Text Header structured field:

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

Additional Content Description

The content of the Outbound Text Header structured field is further described as follows:

OP-TYPE If EW is specified, the following occurs:

- The current buffer address is reset to zero, so subsequent text data is loaded at the start of the buffer.
- The text buffer is cleared.
- The MDT is reset.
- The window origin (row,column) is reset to (1,1).
- If Write is specified, the current buffer address is unchanged, so subsequent text data is appended to existing data.

WCC If the WCC specifies reset MDT, the MDT is reset.

LVL If LVL is X'FF', the LVL value is ignored. Otherwise, LVL identifies the element at which the cursor is to be placed (the first element in any position is at level 1).

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

If there is no element at the specified level, the highest level element at that (row, column) position is assumed.

CRO If CRO is X'FFFF', the cursor row is not changed. Otherwise, the cursor is placed in the specified row (CRO+1), of the presentation space.

CC If CC is X'FFFF', the cursor column is not changed. Otherwise, the cursor is placed in the specified column (CC+1), of the presentation space.

HDR If the HDR data is present, this is processed; otherwise, the existing environment is unchanged.

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

A subsequent Type 1 Text structured field can contain an insert cursor control. This then overrides the cursor position defined by Outbound Text Header.

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

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

Outbound 3270DS

This section describes the Outbound 3270DS structured field.

Function

The Outbound 3270DS structured field directs 3270 data stream orders and data to a named partition.

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

Format

The following table shows the format of the Outbound 3270DS structured field: rules=horiz split=yes scale='.9'. rules=none split=yes scale='.9' arrange='1 2 3 / 4 5 6' expand mindepth=p5.

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

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

Additional Content Description

The content of the Outbound 3270DS structured field is further described as follows:

PID Specifies to which partition the data in this structured field is to be directed. The specified partition must be defined or a data stream error occurs. (See Table A-1 on page A-1.) On a device that does not support multiple partitions, partition 0 must always be specified.

CMND Contains control information for the partition. When CMND = X'6F', no WCC appears in the structured field. CMND values not shown are reserved and are rejected (sense code X'1003' or X'1001').

If the CMND byte specifies BSC Copy, the copy is valid only if the following are true:

- The *from* device and the *to* device are in implicit partition state.
- The communications must be remote BSC.
- The Outbound 3270DS structured field carrying the BSC Copy command must be the last structured field in the transmission.

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

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

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

Present Absolute Format

This section describes the Present Absolute Format outbound structured field.

Function

The Present Absolute Format structured field invokes a *named format* and presents it to a device.

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

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

Format

The following table shows the format of the Present Absolute Format structured field:

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

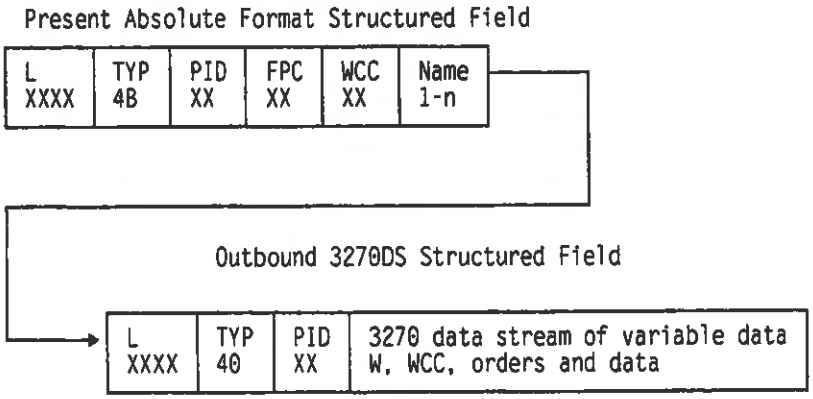
Notes:

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

Additional Content Description

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

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



Data Stream

The relationship between the Format Presentation Command and the 3270DS Outbound structured field command is as follows:

- The first operation performed is that specified by the Present Absolute Format structured field. The command is taken from the Format Presentation Command subfield, the WCC is taken from the WCC subfield, and the orders and data used are retrieved from storage in the control unit based on the format name specified in the Name subfield.
- The second operation performed is that specified by the Outbound 3270DS structured field. The command is taken from byte 4, the WCC is taken from byte 5, and the orders and data from bytes 6 through *n* of the structured field.

Example 1: To erase the screen, present a format, and to merge variable data, an EW command should be specified as the Format Presentation Command in the Present Absolute Format structured field. A Write command should be specified in byte 4 of the Outbound 3270DS structured field to write the variable data to the screen.

Example 2: To present a format without erasing the screen and to merge variable data, a Write command should be specified as the Format Presentation Command in the Present Absolute Format structured field. A Write command should be specified in byte 4 of the Outbound 3270DS structured field to write the variable data to the screen.

Present Relative Format

This section describes the Present Relative Format outbound structured field.

Function

The Present Relative Format structured field invokes a *named format* and increments the buffer addresses contained in the format data stream by the 16-bit binary offset value, specified in the Format Offset Value subfield, before processing the format data for presentation to a device.

To present a format that requires relocation, the Present Relative Format structured field must be used. This structured field functions like the Present Absolute Format structured field, except the buffer addresses contained in the named format are incremented by the 16-bit binary value, before processing the format data stream for presentation to a device. If a Present Format request is made before selecting a format group for the partition, a Format Group Not Selected error occurs, and the Present Format request is rejected.

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

Format

The following table shows the format of the Present Relative Format structured field:

Byte	Content	Content Description
0-1	L	Length of this structured field
2	SFID	X'4C' Identifies this structured field as Present Relative Format
3	PID	Partition identifier (X'00'-X'7E')
4-5	FOV	Format Offset Value: 16-bit binary offset value that is added to the buffer addresses contained in the named format prior to the data stream processing.
6	FPC	Format Presentation Command: X'F1' Write - The data associated with the named format is written to the buffer of the specified partition. X'F5' EW - The viewport and buffer of the specified partition are cleared, and the data associated with the named format is written to the buffer of the specified partition. X'7E' Same action as an EW.
7	WCC	3270 WCC Byte
8-n	NAME	Name of the format being invoked.

Note: If PID is a text partition, an error occurs.

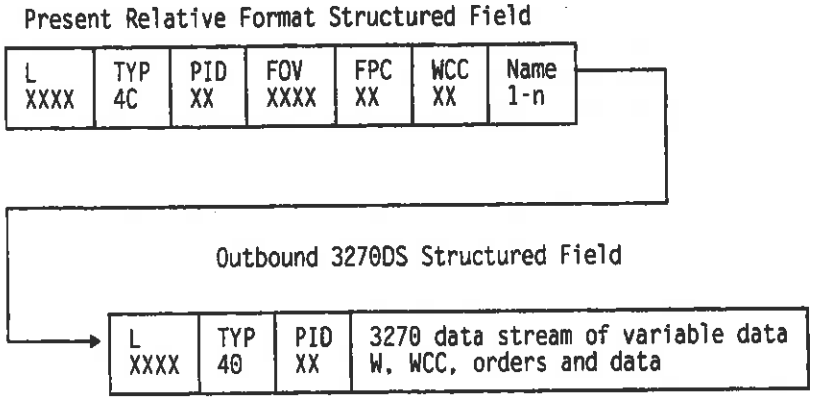
Additional Content Description

The content of the Present Relative Format structured field is further described as follows:

FPC Values of the Format Presentation Command not shown are reserved. If transmitted, they are rejected.

WCC WCC bits 0, 2 and 3 are ignored if transmitted to a display. If WCC bit 1 = 1, a reset is performed. If PID is not equal to zero, only the indicated partition is reset.

To merge variable data with a relocated format, a sequence of Present Relative Format and Outbound 3270DS structured fields must be used, as follows:



The structured fields are processed in the sequence in which they are transmitted in the data stream. Since structured fields can be used in various combinations, each structured field is processed as a separate entity. Using sequences of structured fields in the data stream allows multiple formats to be invoked and merged with multiple streams of variable data within one WSF transmission. The Format Offset Value has no effect on the Outbound 3270DS structured field.

Read Partition

This section describes the Read Partition outbound structured field.

Function

The Read Partition structured field reads data from a specific partition or it queries the device.

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

The Read Partition structured field provides a means for the host to explicitly read a partition. In addition, the host can use the Read Partition structured field to request a 3270 data stream device or workstation to report functions supported by the device or workstation. This information is returned to the host in the form of Query Reply structured fields.

The reply to Read Partition (Query) is sent as a sequence of Query Reply structured fields, as follows:

Format

The following table shows the format of the Read Partition structured field:

Byte	Bit	Content	Content Description
0-1		**L**	Length of this structured field
2		**SFID**	**X'01'** Identifies this structured field as Read Partition
3		**PID**	Partition identifier: **X'00' - X'7E'** (read operations), or **X'FF'** (query operations)
4		**TYPE**	The type of operation to be performed: **X'02'** Query **X'03'** Query List **X'6E'** Read Modified All (RMA) **X'F2'** Read Buffer (RB) **X'F6'** Read Modified (RM)
5	0-1	**REQTYP**	Request Type - present only for Type = X'03' (QUERY LIST) **B'00'** QCODE List **B'01'** Equivalent + QCODE List **B'10'** All **B'11'** Reserved
	2-7	SFID	Reserved
6-n		**QCODE**	Query Reply Codes for Type X'03'

Additional Content Description

The content of the Read Partition structured field is further described as follows:

TYPE Can contain one of the following values:

X'02' The structured field ends after byte 4.

X'03' Byte 5 is a flag byte. Bytes 6-*n* contain the QCODEs of the Query Replies being requested.

Note: Read type codes not shown here are reserved for future extension, and are rejected.

REQTYP Present only if TYPE = X'03'.

B'00' QCODE List indicates the only Query Replies being requested are those specified in bytes 6-n. If the value is B'00' but no list is present (count field is valid), a Null Query Reply is returned.

B'01' (Equivalent + QCODE List) indicates all the Query Replies that would be sent in response to a Query are sent in addition to those QCODEs (if any) that are specified in the QCODE list (bytes 6-n). No duplicate Query Replies are sent. If the QCODE list requests a Query Reply that would be sent anyway, because of the B'01' flag, the Query Reply is sent only once.

B'10' All indicates that *all* the Query Replies that are supported are sent. If a QCODE list is present (bytes 6-n), the All flag overrides the list; that is, the list is ignored.

QCODES Contains the Query Reply codes requested when the TYPE value is X'03'.

The same QCODE can appear more than once in the list (bytes 6-n). However, only one Query Reply is returned for a particular QCODE value regardless of how many times it appears in the list.

All QCODE values in the list are valid. Those QCODEs not supported are ignored. However, if none of the QCODEs in the list are supported, a null Query Reply is returned.

If any of the following conditions exist, Read Partition is rejected (see Table A-1 on page A-1):

- The device is in retry state (sense code X'0871').
- The terminal is not in receive (RCV) or contention (CONT) state.
- Read Partition is not the last structured field in the RU chain.
- The RU chain does not specify CD.
- The operation type is Query or Query List and the PID is not X'FF'.
- The operation type is Read Modified, Read Modified All, or Read Buffer, and PID does not exist.
- The operation type is Read Modified All or Read Buffer, and PID specifies a text partition.

Otherwise, the following steps are performed:

1. The enter-inhibit condition is raised.
2. If the PID specified in Read Partition is not X'FF', INPID is set to the specified PID.
3. INOP is set to the type of operation specified in Read Partition Type Field.
4. For Read Modified (type code X'F6'), Read Modified All (type code X'6E'), or Read Buffer (type code X'F2') the data is transmitted inbound in the format specified by the operation and the inbound reply mode with an AID of X'61'.
5. For a Query (type code X'02'), or Query List (type code X'03') operation, a set of Query Replies is transmitted inbound. These replies describe the features on that device.
6. The display is placed in RCV state.
7. The device is placed in retry read state.

Request Recovery Data

This section describes the Request Recovery Data outbound structured field.

Function

The Request Recovery Data structured field requests recovery data for Print Job Restart when sent from PLU to SLU.

Format

The following table shows the format of the Request Recovery Data structured field:

Byte	Content	Content Description	
0 - 1	L	X'0005'	Length of this structured field
2 - 3	SFID	X'1030'	Identifies this structured field as Request Recovery Data
4	SFID	Reserved	

This structured field must flow to enable the SLU to send the recovery data to the PLU.

Reset Partition

This section describes the Reset Partition outbound structured field.

Function

The Reset Partition structured field resets definable partition characteristics for the specified partition (PID) to their default values:

Partition Characteristic	Default
Inbound Reply mode	Field
Format Group Selection	No format group selected
Set Partition Characteristics	None defined.

Reset partition has no effect if the partition A-MODE is text.

Format

The following table shows the format of the Reset Partition structured field:

Byte	Content	Content Description
0-1	**L**	**X'0004'** Length of this structured field
2	**SFID**	**X'00'** Identifies this structured field as Reset Partition
3	**PID**	Partition identifier (X'00' - X'7E')

Restart

This section describes the Restart outbound structured field.

Function

The Restart structured field indicates that the restart is in progress and that a certain number of pages and lines should be bypassed before printing starts again using the data that follows. This structured field is sent from the PLU to the SLU.

Format

The following table shows the format of the Restart structured field:

Byte	Content	Content Description
0 - 1	**L**	Length of this structured field
2 - 3	**SFID**	**X'1033'** Identifies this structured field as Restart
4	**RES**	Reserved
5 - 6	**START PAGE**	Number of pages to skip on restart
7 - 8	**START LINE**	Number of lines to skip on page for restart
9 - N	**SCS DATA**	SCS data (noncompressed and noncompacted) to set up for restart

The SNA Character String (SCS) data field must include the required Set Horizontal Format (SHF) and/or Set Vertical Format (SVF) and other SCS data for restart. The first byte of Function Management (FM) data resumes at the checkpoint spot or at the start of the SCB string or structured field containing the checkpoint spot in the following RU chain. There are three other data fields:

START PAGE Start Page is the indication sent by the PLU to the SLU of the number of pages that are to be bypassed prior to printing during a restart operation.

START LINE Start Line is the indication sent by the PLU to the SLU of the number of lines on the starting page that are to be bypassed prior to printing during a restart operation.

SCS DATA Data is sent to re-establish various parameters to the state at the time of the checkpoint (for example, the SVF and SHF codes indicated by the vertical and horizontal offsets in the Recovery Data structured field). Any SCS codes (control or graphic) can be included. Counts are reset to those at the time of the checkpoint after processing the SCS codes within the structured field.

SCS Data

This section describes the SCS Data outbound structured field.

Function

The SCS Data structured field intermixes SCS data with other structured fields. The SCS data portion of the SCS Data structured field is treated in the same way as SCS data not delimited by an SCS Data structured field.

Format

The following table shows the format of the SCS Data structured field:

Byte	Content	Content Description
0-1	**L**	Length of this structured field
2	**SFID**	**X'41'** Identifies this structured field as SCS Data
3	**PID**	Partition identifier (X'00' - X'7E')
4-n	**DATA**	SCS data

Select Color Table

This section describes the Select Color Table outbound structured field.

Function

The Select Color Table structured field selects either the nonloadable or the loadable color table.

Format

The following table shows the format of the Select Color Table structured field:

Byte	Content	Content Description
0 - 1	**L**	**X'0006'** Length of this structured field
2 - 3	**SFID**	**X'0F04'** Identifies this structured field as Select Color Table
4 - 5	**COMMAND**	Command

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

Select Format Group

This section describes the Select Format outbound structured field.

Function

The Select Format Group structured field selects the format group that is to be the source of formats in subsequent operations.

The Select Format Group structured field defines, for the specified partition, a single format group that is to be the current source of formats for subsequent Present Absolute and Present Relative Format requests for that partition. The last group selected remains current until it is reset or replaced by a subsequent Select Format Group for that partition. Subsequent Present Absolute and Present Relative Format requests specifying a particular partition are qualified by the current format group for that partition.

If a Present Absolute or Present Relative Format request is made before a format group is selected, it is rejected.

Note: If PID is a text partition the structured field is rejected.

Format

The following table shows the format of the Select Format structured field:

Byte	Content	Content Description
0-1	**L**	Length of this structured field
2	**SFID**	**X'4A'** Identifies this structured field as Select Format Group
3	**PID**	Partition identifier (X'00' - X'7E')
4-n	**GRP**	Format group

Set Checkpoint Interval

This section describes the Set Checkpoint Interval outbound structured field.

Function

The Set Checkpoint Interval structured field passes from the host to the terminal the number of pages that are to be in the interval between checkpoints. It contains the recovery data needed by the host to recover from the error.

This structured field resets all previous checkpointed information. The checkpoint counters begin with the first FM data byte following the structured field.

Format

The following table shows the format of the Set Checkpoint Interval structured field:

Byte	Content	Content Description
0-1	**L**	**X'0007'** Length of this structured field
2-3	**SFID**	**X'1032'** Identifies this structured field as Request Recovery Data
4	**SFID**	Reserved
5-6	**INTERVAL**	Checkpoint interval — Specifies the number of pages in the interval between terminal checkpoints. This number enables checkpointing of jobs that use the short forms without taking an excessive number of checkpoints. A zero value indicates that checkpoints are not to be taken.

Set MSR Control

This section describes the Set MSR Control outbound structured field.

Function

The Set MSR Control structured field allows the application program associated with the host to set the device input states and their associated indicators. In addition, this structured field allows the application program to set user indicators.

Format

The following table shows the format of the Set MSR Control structured field:

Byte	Bit	Content	Content Description
0-1		**L**	**X'000A'** Length of this structured field
2-3		**SFID**	**X'0F01'** Identifies this structured field as Set MSR Control
4		**PID**	Partition identifier
5		**TYPE**	MSR type = X'01' All other values reserved
6		**STATE MASK**	Bit i=B'1' selects states to be modified by corresponding bit in byte 7:
	0	SFID	**USER** User mode
	1	SFID	**LOCK** Locked /Unlocked
	2	SFID	**AUTO** Auto Enter
	3	SFID	**AI1S** Audible Ind 1 Suppress
	4	SFID	**AI2S** Audible Ind 2 Suppress
	5-7	SFID	**RES** Reserved
7	0-7	**STATE VALUE**	**Bit i=B'0'** sets state off **Bit i=B'1'** sets state on
8		**IND MASK**	**Bit j=B'1'** selects corresponding indicator
9		**IND VALUE**	**Bit j=B'0'** sets indicator off **Bit j=B'1'** sets indicator on

Additional Content Description

This structured field is sent by the application to control the following device states and indicators.

Mask Byte 6 (bits 0–4)

Identifies the state of the device. When these bits are set to B'1' the corresponding bit in byte 7 determines whether the state is on or off.

USER This state of the device is set by the application, and the meaning is determined by the application.

LOCK When this state is set, the device is disabled for input.

AUTO ENTER
When this state is set, an input operation on the device will cause an Enter operation from the device, regardless of the type of card read. When this state is reset, an input operation on the device will not cause an Enter operation from the device unless a secure card is read.

AI1S When the AUDIBLE IND 1 SUPPRESS state is set, the internally generated audible indication 1 is suppressed.

AI2S When the AUDIBLE IND 2 SUPPRESS state is set, the internally generated audible indication 2 is suppressed.

STATE VALUE
Determines whether the state is on or off. If the corresponding bit in byte 6 is set to B'1', apply the bit in byte 7.

IND MASK and IND VALUE
These indicators are set by the application program to convey information to the operator of the device. For the Set MSR Control structured field Type X'01', magnetic reader indicators 0–4 are defined. These indicators are reset by the next read to type the correct partition. The device states set by the Set MSR Control structured field are device related, while the magnetic reader indicators are partition related. Therefore, the magnetic reader indicators are saved on a partition basis and only affect the device when the partition is active.

Set Partition Characteristics

This section describes the Set Partition Characteristics outbound structured field.

Function

The Set Partition Characteristics structured field provides a base for optional parameters which further define the partition characteristics beyond those given in the Create Partition structured field.

Format

The following table shows the format of the Set Partition Characteristics structured field:

Byte	Bit	Content	Content Description
0-1		L	Length of this structure
2-3		SFID	X'0F08' Identifies this structured field as Set Partition Characteristics
4		PID	Partition identifier (X'00' - X'7E' or X'FF')
5-6		FLAGS	Reserved

Operation

The Set Partition Characteristics structured field is used to change operational characteristics at a logical terminal. The characteristics which are subject to change must have default values or definitions at logical terminal initialization (power on) or reset time. These definitions are how the characteristics would behave in a 3270 subsystem that did not allow for changing of the characteristics and are referred to as *3270 Default Function definitions.*

When the Set Partition Characteristics structured field is allowed to change the operational characteristics, the changes can be directed to apply to all partitions or specific partitions. When the changes are to apply to all partitions, they are referred to as *Logical Terminal Characteristics.* When the changes are to apply to a specific partition, they are referred to as *Partition Characteristics.*

An inheritance structure exists between the 3270 defaults, the logical terminal, and the partitions. This inheritance is for the partition to inherit from the 3270 default definition. Thus, without any Set Partition Characteristics structured field, all partitions would behave as a logical terminal which did not support the structured field. When a Set Partition Characteristics structured field is sent with a PID of X'FF', the accompanying SDPs set the logical terminal characteristics. Now it is these settings instead of the 3270 default functions which are inherited by present or future partitions created in the logical terminal. These functional definitions for a given self-defining parameter apply until the logical terminal characteristic is altered or the logical terminal is reset.

Finally, when the Set Partition Characteristics SDPs are directed to the specific partition, the characteristics are set for that partition only. Partition Characteristics take precedence over the Logical Terminal Characteristics and the 3270 Default Function Definition for a given self-defining parameter. These functional definitions for a given self-defining parameter apply until the partition characteristic is altered or the partition is reset.

This structured field can be sent any number of times to change the characteristics of the addressed partition (Partition ID of X'00' – X'7E') or the logical terminal (Partition ID of X'FF').

A partition characteristic established by this structured field is retained until the partition no longer exists (see reset information following) or until the characteristic is changed by another occurrence of this structured field having a self-defining parameter which alters it.

If no self-defining parameters are included in this structured field, then the Partition Characteristics structured field has no effect.

An error situation exists when any of the following arises:

- The structured field is not supported.
- The named partition does not exist.
- The self-defining parameter is not supported.

- The Set Partition Characteristics structured field is addressed to the logical terminal (PID = X'FF'), but the Partition Characteristics Query Reply does not indicate logical terminal support of an included self-defining parameter.
- The Set Partition Characteristics structured field is addressed to a partition (PID = X'00' – X'7E'), but the Partition Characteristics Query Reply does not indicate partition support of an included self-defining parameter.

The Set Partition Characteristics structured field can include one or more self-defining parameters. The self-defining parameters provide a functional definition of their respective operations. At the logical terminal level, the self-defining parameters can specify that:

- The 3270 Default Function Definition is to be inherited.
- The functional definition is the same as the 3270 Default Function Definition but is not inherited.
- That one of the alternative functional definitions is to apply.

For a specific partition, the self-defining parameters can specify that:

- The Logical Terminal Characteristics functional definition is to be inherited.
- The functional definition is the same as the Logical Terminal Characteristics but is not inherited.
- One of the alternative functional definitions is to apply.

The self-defining parameters can be sent as a 2-byte structure, length byte and self-defining parameter identifier only, or as a structure containing more than two bytes when the parameter values are included. A self-defining parameter more than two bytes long explicitly provides a functional definition at the addressed level (logical terminal or partition) even though this functional definition can be synonymous with a functional definition at a lower level.

When a self-defining parameter is more than two bytes long, but less than its maximum potential length, the omitted parameters assume defined values to complete the functional definition at the addressed level. When a 2-byte self-defining parameter is sent (length byte and identifier byte only), the functional definition at the addressed level (logical terminal or partition) is set to inherit.

Parameters set by a self-defining parameter addressed to the logical terminal are reset to inherit by the following:

- A Set Partition Characteristics structured field addressed to the logical terminal and including a self-defining parameter that includes only the length byte and the identifier byte.
- An Erase/Reset structured field, a WCC reset bit on with Erase/Write or Erase/Write Alternate command, Power off, Bind, Clear local function, and System Request local function. Note that destroying the last partition does not reset the parameters.

Parameters set by a self-defining parameter addressed to an individual partition are reset to inherit when:

- A Set Partition Characteristics structured field is addressed to that partition and includes a self-defining parameter that includes only the length byte and the identifier byte.
- An individual partition is reset by Partition Reset, Erase/Reset, Create Partition for a partition with the same PID, Destroy Partition, WCC reset bit on with Erase/Write or Erase/Write Alternate in command form or in an Outbound 3270DS structured field.

Viewport Outline Self-Defining Parameter

The Viewport Outline self-defining parameter specifies the maximum thickness of the distinguishing outline of a viewport. The outline of the viewport begins at the outermost edge of the outermost row or column of the viewport. This outline consumes up to Byte 2 pel positions from the outermost edge of the viewport. The outline thickness is specified by the outline thickness value.

The following table shows the format of the Viewport Outline self-defining parameter:

Byte	Content	Content Description	
0	**L**	**X'06'**	Length of this structure
1	**SDPID**	**X'01'**	Identifies this self-defining parameter as Viewport Outline Parameters
2	**OT**	Top edge outline thickness	
3	**OB**	Bottom edge outline thickness	
4	**OL**	Left edge outline thickness	
5	**OR**	Right edge outline thickness	

The values specified in bytes 2 through 5 of the Viewport Outline self-defining parameters range in value from zero to the value specified in the Viewport Outline structured field of the Partitions Characteristics Query Reply.

An error exists when any of the following arises:

- The self-defining parameter specifies an incorrect length.
- A self-defining value is outside the range specified as supported by the Query Reply.

Enable User Call-up Self-Defining Parameter

This Set Partition Characteristics self-defining parameter is used to provide for the selection of the operational mode of User Selected Formats.

The following table shows the format of the Enable User Call-up self-defining parameter:

Byte	Bit	Content	Content Description	
0		L	X'03'	Length of structure
1		SDPID	X'04'	Identifies this self-defining parameter as Local User Call-up
2		FLAGS	5	
	0	ENABLE	User Selectable Formats	
			B'0'	Disable User Selectable Formats
			B'1'	Enable User Selectable Formats
	1	SEARCH	Perform name search	
			B'0'	Perform name search globally
			B'1'	Perform name search qualified by the currently selected group name.
	2-7	RES	Must be set to zeros	

The content of the Enable User Call-up self-defining parameter is further described as follows:

ENABLE User selected formats are enabled if the SDP is received with this bit set to B'1'. If the bit is set to B'0', they are disabled.

SEARCH If this bit is set to B'0', the local name search is performed for all stored formats. If this bit is set to B'1', the local name search is performed only for formats assigned to the currently selected group.

Operation

These parameters remain set until one of the following conditions occurs:

- A Set Partition Characteristics structured field is received that resets the parameter.
- An Erase/Write or Erase/Write Alternate command is received with WCC = Reset.
- An SNA bind is received.
- An Erase/Reset structured field is received.

Note: These parameters are not reset by the Clear key, SYS REQ key, TEST key, or by turning on the power to the device.

Select Base Character Set Self-Defining Parameter

This Set Partition Characteristics self-defining parameter is used to select the Base Character Set that is to be used at the device.

The following table shows the format of the Select Base Character Set self-defining parameter:

Byte	Bit	Content	Content Description	
0		L	X'04'	Length of structure
1		SDPID	X'05'	Identifies this self-defining parameter as Select Base Character Set
2		FLAGS1		
	0-3	BASE	Base Character Set Index	
	4-7		Reserved - must be set to zeros	
3		FLAGS2	Reserved - must be set to zeros	

The content of the Select Base Character Set self-defining parameter is further described as follows:

BASE The Base Character Set is established by indexing the list of Base Character Set descriptors in the order returned in the Character Sets Query Reply.

Set Printer Characteristics

This section describes the Set Printer Characteristics outbound structured field.

Function

The Set Printer Characteristics (SPC) structured field allows the application to control the setting and resetting of certain functions and modes within a control unit terminal (CUT) printer. See "Early Print Complete Self-Defining Parameter" on page 5-67.

The SPC structured field is valid only in non-SNA environments.

Format

The following table shows the format of the Set Printer Characteristics structured field:

Byte	Bit	Content	Content Description
0-1		**L**	Length of this structured field
2-3		**SFID**	**X'0F84'** Identifies this structured field as Set Printer Characteristics
4		**FLAGS**	
	0	RSTALL	Reset All Characteristics **B'0'** Off (No reset all) **B'1'** On (Reset all)
	1-7	Reserved	Reserved
5		**FLAGS**	Reserved

The base can be sent without any self-defining parameters appended when a reset of all previously set functions is desired without setting any new ones (RSTALL=ON). If the base is sent without any self-defining parameters appended with No Reset All indicated (RSTALL = OFF), it is treated as a No-op by the printer and a normal response is returned.

Within a WSF transmission, the SPC must be sent as the first structured field in the transmission. However, if the SPC structured field is used with the Data Chain structured field, it must be sent immediately following a Data Chain (Begin or Only) structured field. If it is used after a Data Chain (Begin) structured field, and a recoverable error occurs before the end-of-chain, the unit of recovery is the chain. An example of recovery would be to retry the chain from the beginning.

The Early Print Complete (EPC) self-defining parameter is used to set an early print complete mode of operation on or off. This parameter provides for enabling/disabling of the printer operator control. (EPC mode can be on or off depending on the setting of the operator control.) The default is printer operator control enabled.

If the RSTALL flag in the base part of the structured field equals Reset, the default is set. However, when the Early Print Complete self-defining parameter is present, the SREPC flag overrides the RSTALL flag.

For example, if RSTALL = Reset and SREPC = EPC on (operator control disabled), the EPC mode is on and operator control is disabled.

Early Print Complete Self-Defining Parameter

The Early Print Complete (EPC) self-defining parameter can be appended to the base portion of this structured field. EPC allows the loading of the printer buffer and the printing operation to overlap, thereby improving throughput to high-speed printers. EPC is supported only in non-SNA environments. See "Early Print Complete (EPC) Operations" on page 8-13.

The following table shows the format of the Early Print Complete self-defining parameter:

Byte	Bit	Content	Content Description	
0		L	'03'	Length of parameter
1		SDPID	X'01'	Identifies this self-defining parameter as Early Print Complete
2		SREPC	Set/Reset Early Print Complete	
	0-1	SFID	B'00'	Enable operator control
			B'01'	EPC Off – disable operator control
			B'10'	EPC On – disable operator control
			B'11'	Reserved
	2-7	Reserved	Reserved	

Prior to use of the SPC structured field, an application should determine the characteristics of the printer by issuing a Read Partition Query List structured field. The Query List structured field must either contain a "Q" code of X'A9' or indicate ALL.

When EPC is on, the printer returns a positive completion status for the print information received after it is verified but before it is printed. If an error occurs while this information is printing, the error is reported and associated with print information received later. Because the error is not properly synchronized with print information to which it pertains, error handling can be affected.

If an SPC structured field is detected out of sequence, it is rejected with an Op Check sense code sent to the host.

All set printer characteristics are reset to default by any of the following:

- Receipt of an EW or EWA command with WCC = Reset
- Power-on reset (POR)
- RSTALL=X'1B'.

If a system performs its error handling by reprinting the job, EPC should be turned on at the beginning of the job and turned off with the last piece of print information for the job. This requires the code that manages EPC to recognize job boundaries. In the SNA environment, LU1 can be used to overlap the loading of a printer's buffer with the print operation if the serial load and print required by LU3 results in performance problems.

Set Reply Mode

This section describes the Set Reply Mode outbound structured field.

Function

The Set Reply Mode structured field defines the format of any subsequent inbound data streams from an alphanumeric partition by defining the values required by the host application.

The Reply Modes Query Reply is used to inform the host of the attribute orders that the device supports in both the inbound and the outbound direction and the modes that can be set by the Set Reply Modes structured field.

The Set Reply Mode structured field is sent outbound to devices that support at least the Extended Field Mode of operation. The Extended Field Mode indicates support of the Start Field Extended (SFE) and the Modify Field (MF) orders. See "Start Field Extended (SFE)" on page 4-4 and "Modify Field (MF)" on page 4-7. Character Mode can be selected if the device has indicated its support in the Reply Modes Query Reply. See "Set Attribute (SA)" on page 4-6.

Subsequent inbound data streams from the named partition sent inbound in response to a read command are in the format defined by the Set Reply Mode structured field. This reply mode remains in effect until set by another Set Reply Mode or until reset by a reset action.

The inbound data stream from a Text partition is not controlled by a Reply Mode. Hence, a Set Reply Mode structured field sent to a text partition is rejected.

Format

The following table shows the format of the Set Reply Mode structured field:

Byte	Content	Content Description
0-1	**L**	Length of this structured field
2	**SFID**	**X'09'** Identifies this structured field as Set Reply Mode
3	**PID**	Partition identifier (X'00' through X'7F')
4	**MODE**	**X'00'** Field mode **X'01'** Extended field mode **X'02'** Character mode **Other** Reserved
5-n	**ATTR LIST (Present only if mode is X'02')**	The type codes for the attribute types that: • Are transmitted using SA orders if mode X'02' is specified. • The operator can select from the keyboard. (The operator can select the attribute value to associate with the keyed data.)

Additional Content Description

The values that can appear in each mode are listed below:

Mode	Values
Field	3270 orders (SBA, SF) Characters Field attributes GE
Extended field	3270 orders (MF, SBA, SF, SFE) Characters Field attributes Extended field attributes GE
Character	3270 orders (MF, SBA, SF, SFE, SA) Characters Field attributes Extended field attributes Character attributes listed in Set Reply Mode structured field GE.

In field mode or extended field mode, attribute selection by the operator is not allowed. Any operator selection allowed by a previous Set Reply Mode structured field is reset to the default value.

In character mode, the listed attribute types can have values selected by the operator. Any operator selection current when Set Reply Mode is interpreted is not reset. Attribute selection by the operator for attribute types not explicitly listed is not allowed.

Mode values not shown here are reserved and are rejected. Attribute types other than those defined are rejected.

Set Window Origin

This section describes the Set Window Origin outbound structured field.

Function

The Set Window Origin structured field changes the position of the window within the presentation space of the specified partition.

If the named partition does not exist, an error situation arises. (See Table A-1 on page A-1.)

The window of the specified partition is repositioned to the new origin. If the following condition occurs, the window is not moved and a sense code is returned:

RW + HW > H
CW + WW > W

where:

HW Height of window (viewport)
H Height of presentation space
WW Width of window (viewport)
W Width of presentation space.

The current cursor position for the named partition is not changed unless the new window origin results in the current cursor position's being outside the window. If outside the window, the current cursor position is moved by the minimum number of row and columns needed to place it in the nearest row or column of the window.

Note: The Modify Partition structured field can also be used to change the window position.

Format

The following table shows the format of the Set Window Origin structured field:

Byte	Content	Content Description
0-1	**L**	**X'0008'** Length of this structured field
2	**SFID**	**X'0B'** Identifies this structured field as Set Window Origin
3	**PID**	Partition identifier (X'00' through X'7E')
4-5	**RW**	Row position of the window origin
6-7	**CW**	Column offset of the window origin

Type 1 Text Outbound

This section describes the Type 1 Text Outbound structured field.

Function

The Type 1 Text Outbound structured field transmits data to a text partition.

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

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

Format

The following table shows the format of the Type 1 Text Outbound structured field:

Byte	Content	Content Description
0-1	**L**	Length of this structured field
2-3	**SFID**	**X'0FC1'** Identifies this structured field as Type 1 Text Outbound
4	**PID**	Partition identifier
5-6	**RES**	Reserved
7-n	**DATA**	Text data

Additional Content Description

The content of the Type 1 Text Outbound structured field is further described as follows:

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

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

- An unsupported X'2B' control is detected.
- A parameter error occurs in a X'2B' control.
- An unsupported 1-byte control is detected.
- The text data cannot be contained in the text buffer.

During this interpretation, nulls (X'00') are treated as No-op. They are not placed in the text buffer.

As the text data is interpreted, the current buffer address is incremented by one for each byte placed in the buffer. The current buffer address is updated both by host writes, and by operator interaction (that is, it is always the address of the first empty byte in the buffer).

The buffer is formatted to the presentation space.

If DATA contains an insert cursor control, the screen cursor is placed at the presentation position of the element immediately following this insert cursor control. If there are no elements in DATA following the insert cursor, the screen cursor is placed at the presentation position immediately following that of the last element of DATA.

The window and cursor positions are then resolved as follows:

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

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

Outbound/Inbound Structured Fields

The structured fields described in the following sections are structured fields that can be transmitted either from the application to the device or from the device to the application. The structured field format is the same for both inbound and outbound structured fields.

Data Chain

This section describes the Data Chain outbound/inbound structured field.

Function

The Data Chain structured field provides a data-chaining function for non-SNA environments. It identifies the individual transmissions as being part of a single message.

This structured field uses the structured field grouping function described in "Structured Field Grouping" on page 5-6 to provide a chaining function. Items such as structured fields and control code sequences contained in the message can span transmissions that are related through Data Chain structured fields. This allows a message to be divided into transmissions without any examination or manipulation of the message content. The Data Chain structured field was defined for use in the non-SNA environment. Its use in the SNA environment is unnecessary because of the chaining function provided in SNA.

Format

The following table shows the format of the Data Chain structured field:

Byte	Bit	Content	Content Description
0-1		L	X'0006' Length of this structured field
2-3		SFID	X'0F21' Identifies this structured field as Data Chain
4	0	RES	Reserved
	1-2	GROUP	B'00' Continue B'01' End B'10' Begin B'11' Only
	3-4	INCTRL	Inbound Control B'00' No change B'01' Enable Inbound Data Chaining B'10' Disable Inbound Data Chaining B'11' Reserved
	5-7	RES	Reserved
5		FLAGS	Reserved

Additional Content Description

The content of the Data Chain structured field is further described as follows:

GROUP Establishes whether the data following the Data Chain is the beginning, continuation, or the end of a chain of grouped transmissions. The value B'11' (only) indicates the entire message is contained in the transmission that follows the Data Chain structured field.

INCTRL This parameter is only set in the outbound Data Chain structured field to control how the inbound data is sent. It is not set in the inbound Data Chain structured field and should be zeros.

An outbound Data Chain structured field with INCTRL = B'01' (Enable Inbound Data Chaining) allows the device to use the Data Chain structured field for inbound transmissions. If inbound data chaining is already enabled, it remains enabled.

An outbound Data Chain structured field with INCTRL = B'10' (Disable Inbound Data Chaining) prevents the device from using the Data Chain structured field for inbound transmissions. If inbound data chaining is already disabled, it remains disabled until subsequently enabled by an outbound Data Chain structured field with INCTRL = B'01'. (Enable Inbound Data Chaining).

An outbound Data Chain structure field with INCTRL = B'00' (no change) has no effect on the enable/disabled state.

Destination/Origin

This section describes the Destination/Origin outbound/inbound structured field.

Function

The Destination/Origin structured field identifies the destination or origin of structured fields in a single session multi-device (workstation) implementation.

Outbound (from the host) the ID indicates the destination of the structured fields that follow. Inbound (to the host), the ID indicates the origin of the structured fields that follow.

At the beginning of a transmission, the destination or origin is the default (primary display).

Once a Destination/Origin structured field establishes the destination or origin, it applies until either another Destination/Origin structured field establishes a new destination or origin or a new transmission starts.

Format

The following table shows the format of the Destination/Origin structured field:

Byte	Bit	Content	Content Description
0-1		**L**	**X'0008'** Length of this structured field
2-3		**SFID**	**X'0F02'** Identifies this structured field as Destination/Origin
4		**FLAGS**	
	0-1	INCTRL	Input Control: **B'00'** Enable input **B'01'** No change **B'10'** Disable input **B'11'** Reserved
	2-7	RES	Reserved
5		**FLAGS**	Reserved
6-7		**SFID**	Identifies destination or origin of the structured fields which follow in the data stream

Additional Content Description

The content of the Destination/Origin structured field is further described as follows:

INCTRL Applies only on outbound (to the auxiliary device); on inbound the INCTRL flag is ignored.

B'00' Allows the auxiliary device to send data. If the auxiliary device is already enabled, it remains enabled.

B'01' A change does not occur in the enabled/disabled status.

B'10' The auxiliary device is not permitted to send data until subsequently enabled by a Destination/Origin structured field with INCTRL flag = B'00' - If the auxiliary device is already disabled, the INCTRL flag = B'10' causes no change.

If a Destination/Origin structured field is directed to the base display (ID = X'0000'), the INCTRL flag applies on a global basis: all the auxiliary devices supported are enabled, disabled, or unchanged as a group.

Note: There is one exception where an auxiliary device can send input without being enabled. An Exception Condition structured field, reporting unavailability of the auxiliary device, can be sent in reply to a Destination/Origin structured field sequence attempting to use the auxiliary device.

DOID The valid values for the DOID are:

- X'0000' (permanently assigned to the primary display)
- All ID values returned in the Query Replies for Auxiliary devices.

All other values are invalid and must be rejected.

Object Control

This section describes the Object Control outbound/inbound structured field.

Function

The Object Control structured field carries the Object Data between the device and the host. It is used both inbound and outbound. When used outbound, it indicates the mode of interpretation for the Object Data.

Format

The following table shows the format of the Object Control structured field:

Byte	Bit	Content	Content Description
0-1		**L**	Length of this structured field
2-3		**SFID**	**X'0F11'** Identifies this structured field as Object Control
4		**PID**	Partition identifier
5		**FLAGS**	
	0-1	SPANF	Spanning flag (outbound) **B'00'** Not first and not last (middle) **B'01'** Last but not first **B'10'** First but not last **B'11'** First and last (only)
	2-3	MODE	Control interpretation mode (outbound) **B'00'** Immediate mode **B'10'** Store mode **Other** Reserved
	4-7	RES	Reserved
6		**OBJTYP**	Object type **X'00'** Graphics **X'01'** Image
7-n		**DATA**	Data appropriate to the object type. For the format and contents of this parameter, refer to the appropriate graphics or image publications.

Additional Content Description

The content of the Object Control structured field is further described as follows:

SPANF Permits the segmentation of a structured field into several related structured fields of the same type. The rule for spanning is that the DATA parameters of structured fields of that type are logically contiguous. These indicators are applicable to both inbound and outbound transmissions.

MODE Specifies how the Object Data contained in the DATA parameter (byte 7–n) should be interpreted in the device. This flag is applicable to outbound transmissions only.

- B'00' Immediate mode: For graphics, data units contained within the DATA parameter are stored if the device has this capability; otherwise they are rejected.
- B'10' Store mode: For graphics, data units contained within the DATA parameter are stored if the device has this capability; otherwise it is rejected.

OBJTYP Contains the object type being carried in this structured field.

DATA Contains the self-describing Object Data commands. There can be more than one entity of an object type specified. Entities of different object types must not be mixed in this parameter, since it is interpreted by the object type and therefore correct interpretation of mixed objects cannot be guaranteed.

Object Data

This section describes the Object Data outbound/inbound structured field.

Function

The Object Data structured field carries the object data between the device and the host.

It is used both inbound and outbound. When used outbound, it indicates the mode of interpretation for the object data.

Format

The following table shows the format of the Object Data structured field:

Byte	Bit	Content	Content Description
0-1		**L**	Length of this structured field
2-3		**SFID**	**X'0F0F'** Identifies this structured field as Object Data
4		**PID**	Partition identifier
5		**FLAGS**	
	0-1	SPANF	Spanning flag (outbound) **B'00'** Not first and not last (middle) **B'01'** Last but not first **B'10'** First but not last **B'11'** First and last (only)
	2-3	MODE	Control interpretation mode (outbound) **B'00'** Immediate mode **B'10'** Store mode **Other** Reserved
	4-7	RES	Reserved
6		**OBJTYP**	Object type **X'00'** Graphics **X'01'** Image
7-n		**DATA**	Data appropriate to the object type. For the format and contents of this parameter, refer to the appropriate graphics or image publications.

Additional Content Description

The content of the Object Data structured field is further described as follows:

SPANF Permits the segmentation of a structured field into several related structured fields of the same type. The rule for spanning is that the DATA parameters of structured fields of that type are logically contiguous. These indicators are applicable to both inbound and outbound transmissions.

MODE Specifies how the object data contained in the DATA parameter (byte 7–n) should be interpreted in the device. This flag is applicable to outbound transmissions only.

- B'00' Immediate mode: For graphics, data units contained within the DATA parameter are stored if the device has this capability; otherwise they are rejected.
- B'10' Store mode: For graphics, data units contained within the DATA parameter are stored if the device has this capability; otherwise it is rejected.

OBJTYP Contains the object type being carried in this structured field.

DATA Contains the self-describing Object Data commands. There can be more than one entity of an object type specified. Entities of different object types must not be mixed in this parameter, since it is interpreted by the object type and therefore correct interpretation of mixed objects cannot be guaranteed.

Object Picture

This section describes the Object Picture outbound/inbound structured field.

Function

The Object Picture structured field carries the object pictures between the device and the host. It is used both inbound and outbound. When used outbound, it indicates the mode of interpretation for the object pictures.

Format

The following table shows the format of the Object Picture structured field:

Byte	Bit	Content	Content Description
0-1		**L**	Length of this structured field
2-3		**SFID**	**X'0F10'** Identifies this structured field as Object Picture
4		**PID**	Partition identifier
5		**FLAGS**	
	0-1	SPANF	Spanning flag (outbound) **B'00'** Not first and not last (middle) **B'01'** Last but not first **B'10'** First but not last **B'11'** First and last (only)
	2-3	MODE	Interpretation mode (outbound) **B'00'** Immediate mode **B'10'** Store mode **B'01'** Reserved **B'11'** Store and draw mode
	4-7	RES	Reserved
6		**OBJTYP**	Object type **X'00'** Graphics **X'01'** Image
7-n		**DATA**	Data appropriate to the object type. For the format and contents of this parameter, refer to the appropriate graphics or image publications.

Additional Content Description

The content of the Object Picture structured field is further described as follows:

SPANF Permits the segmentation of a structured field into several related structured fields of the same type. The rule for spanning is that the DATA parameters of structured fields of that type are logically contiguous. These indicators are applicable to both inbound and outbound transmissions.

MODE Specifies how the object pictures contained in the DATA parameter (byte 7–n) should be interpreted in the device. This flag is applicable to outbound transmissions only.

B'00' Immediate mode: For graphics, each chained segment is executed immediately as it is received and not stored.

Non-chained segments and extended drawing routines are stored if the device has this capability; otherwise they are rejected.

B'10' Store Mode: Each object picture contained within the DATA parameter is stored if the device has this capability; otherwise it is rejected. No operation occurs.

B'11' Store and draw mode: For graphics, each segment and extended drawing routine contained within the DATA parameter is stored if the device has this capability; otherwise it is rejected. Additionally, chained segments are executed.

OBJTYP Contains the object type being carried in this structured field.

DATA Contains the self-describing Object Picture commands. There can be more than one entity of an object type specified. Entities of different object types must not be mixed in this parameter, since it is interpreted by the object type and therefore correct interpretation of mixed objects cannot be guaranteed.

OEM Data

This section describes the OEM Data outbound/inbound structured field.

Function

The OEM Data structured field carries the non-IBM-defined data (or a value-added version) to or from another equipment manufacturer's (OEM) auxiliary device. An OEM device is defined here as a device that is manufactured outside of IBM. The device can carry either an outside manufacturer logo or an IBM logo.

The data to or from an auxiliary device on a 3270 data stream workstation must be in structured field form. The OEM Data structured field provides a means of carrying the manufacturer's defined data (or a value added version) to or from an OEM auxiliary device.

Optional Parameters

All parameters through byte 6 must be present. Parameters in bytes 7–n are optional. An OEM Data structured field of length 7 is valid; for example, it could be used to end a group when there is no more data available to send.

Format

The following table shows the format of the OEM Data structured field:

Byte	Bit	Content	Content Description
0-1		**L**	Length of this structured field
2-3		**SFID**	**X'0F1F'** Identifies this structured field as OEM Data
4		**RES**	Reserved
5	0-1	**GROUP**	**B'00'** Continue **B'01'** End **B'10'** Begin **B'11'** Only
	2-7	RES	Reserved for flags
6		**FLAGS**	Reserved
7-n		**DATA**	Optional data

Additional Content Description

The content of the OEM Data structured field is further described as follows:

GROUP Provides a spanning function; that is, it allows data to be divided into two or more OEM Data structured fields without regard to control sequence boundaries.

DATA The OEM Auxiliary Device Query Reply returned for a particular OEM auxiliary device indicates whether the data in the OEM Data structured field contains the manufacturer-defined data stream or a registered value-added version of the manufacturer-defined data stream.

Save/Restore Format

This section describes the Save/Restore outbound/inbound structured field.

Function

The Save/Restore Format structured field is used for saving and restoring a Format Parameter Control Block (FPCB) either at the device (a printer) or at the host location. The FPCB consists of parameters such as top, bottom, left, and right margins. Values are initially set by default values at the device and later modified by SCS control codes received in the data stream.

The function allows the host application to rapidly switch from one set of formats to another. For example, the host application may create parameter set 1 and operate with it for awhile and then save it for later use. It can then create parameter set 2, operate with it as required and then save set 2. Following this, the host application can restore set 1 and continue the operation.

Format

The following table shows the format of the Save/Restore structured field:

Byte	Bit	Content	Content Description	
0-1		**L**	Length of this structured field	
2-3		**SFID**	**X'1034'**	Identifies this structured field as Save/Restore Format
4		**FLAGS**		
	0	SAVE/RES	**B'0'**	Save the current Format Parameter Control Block (FPCB).
			B'1'	Restore saved FPCB.
	1	SEC/PRI	**B'0'**	FPCB saved at secondary
			B'1'	FPCB saved at primary
	2-7	RES	Reserved	
5-n		**FPCB**	Contents of the FPCB that is to be saved or restored	

Additional Content Description

The content of the Save/Restore structured field is further described as follows:

SAVE/RES The SAVE/RES flag determines whether the FPCB is to be saved or restored. If SAVE/RES is set to B'0', the information is sent to FPCB storage. If SAVE/RES is set to B'1', the information is sent from FPCB storage to the device.

SEC/PRI This flag determines the location where the FPCB is to be stored. If the flag is set to B'0', the FPCB is saved in the device and if the flag is set to B'1', the FPCB is sent to the host location for storage.

FPCB The Format Parameter Control Block (FPCB) is a device-dependent control block that contains formatting parameters established by power-on defaults or subsequently received SNA Character String controls such as:

- Set Horizontal Format (SHF)
- Set Vertical Format (SVF)
- Set Line Density (SLD)
- Set Print Density (SPD).

The printer has a current FPCB and one FPCB save area. There are no constraints on the number of save areas at the host location.

The FPCB must contain all of the information necessary to save or restore all format parameters which the printer supports. After a save/restore operation of any given FPCB, either at the printer or the host, the format parameters in effect remain exactly the same as those in effect before the save/restore operation. The save/restore operation itself causes no movement of the current position in the presentation space; that is, the presentation space position is not saved or restored.

At power-on time, the printer initializes the save area to the default values of the device.

Select Intelligent Printer Data Stream (IPDS) Mode

This section describes the Select IPDS Mode outbound/inbound structured field.

Function

The Select IPDS Mode structured field selects the IPDS mode when IPDS is supported in 3270 non-SNA. Refer to your product publications for information on how the data is processed when IPDS is selected.

Format

The following table shows the format of the Select IPDS Mode structured field:

Byte	Content	Content Description
0-1	L	X'0006' Length of this structured field
2-3	SFID	X'0F83' Identifies this structured field as Select IPDS Mode
4-5	FLAGS	Reserved