Personal System/2
XGA Adapter Interface
Technical Reference

15F2154
S15F-2154-0

Personal System/2
XGA Adapter Interface
Technical Reference

First Edition (September 1990)

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

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

Requests for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

This edition applies to the IBM Personal System/2 XGA Adapter Interface.

Special notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

The following terms, denoted by an asterisk (*), used in this publication, are trademarks of the IBM Corporation in the United States or other countries:

IBM Micro Channel Personal System/2
PS/2

The following terms, DENOTED BY A DOUBLE ASTERISK (**) used in this publication, are trademarks of other companies as follows:

Intel

Intel is a trademark of the Intel Corporation.

Preface

This manual is the Technical Reference for the Extended Graphics Array (XGA) Adapter Interface. It is published for the information of applications programmers and others who need to understand the technical details of the XGA Adapter Interface.

The manual is arranged as follows:

- **Chapter 1** describes XGA.
- **Chapter 2** describes the XGA Adapter Interface functions.
- **Chapter 3** details the programming considerations for each order in the Base Function Set; this level of function is common to all the adapters.
- **Chapter 4** details the programming considerations for each order in the Extended Function Sets. Extended Function Set (1) will be supported by all new adapters, but may not be supported by all previous adapters. Extended Function Set (2) will be supported by specific new adapters.
- **Appendix A** discusses applications compatibility between the XGA Adapter Interface and the IBM Display Adapter 8514/A Interface.
- **Appendix B** gives details of the default palettes.
- **Appendix C** gives details of the font file format.

Related Publications

The *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* contains the BIOS interface information.

The *IBM Hardware Interface Technical Reference* contains a description of the basic operation of the video subsystem, including a complete description of the VGA mode.

The *XGA Display Adapter/A Technical Reference* contains a basic operation of the video adapter, including POS information.

Contents

Figures

Chapter 1. XGA

XGA is a hardware feature that provides advanced image and graphics display functions for the IBM Personal System/2. Throughout this Technical Reference, the words "Personal Computer" and the acronym "PC" are used to mean any IBM* Personal System/2* that incorporates XGA and the XGA Adapter Interface, unless specifically stated otherwise. XGA is used to mean the XGA video subsystem incorporated in certain models of the PS/2, and the function provided by the XGA Display Adapter/A. The words "Adapter Interface" are used to mean the programming interface between the XGA and the controlling environment.

The XGA Adapter Interface supports a superset of the adapter interface orders provided by the IBM Personal System/2 Display Adapter 8514/A.

XGA provides a programming interface which effectively automates many of the complex hardware/register linkages that are required by graphics applications. Programming with the XGA Adapter Interface is therefore much easier than programming direct to the PC display hardware. The set of orders accessed by the call interface supports many commonly-required display adapter functions.

Operating Modes

XGA operates in one of two modes, selected under program control:

1. Video-Graphics Array (VGA)
2. Advanced-function.

VGA is the default mode. For further information, see the *IBM Hardware Interface Technical Reference*.

The advanced functions selectable with XGA, are described in Chapter 2.

For information on palette control, see "Palettes" on page 1-2.

XGA supports either 1024 x 768 pel or 640 x 480 pel displays when in advanced function mode.

* Trademark of the IBM Corporation. For a list of trademarks, see page iii.

Color and Gray Scale Ranges

XGA can be used to display:

- Sixteen or 256 colors from a total of 256K (K equals 1024), according to the number of bit planes installed.
- Multiple colors selected from a loadable palette; this option allows:
  - Use of application-specific colors
  - Use of the palette to highlight or otherwise control the visibility of different data areas in the display, without changing the values in the bit planes.
- Sixteen or 64 levels of gray scale, according to the number of bit planes installed; this option supports "image" and "anti-aliased" character sets.

Bit Plane Storage

Display buffer storage is arranged in planes of 1024 x 1024 bits by default. The size of the current display bitmap may be set (in advanced-function mode) by the HSBMAP order; in VGA mode, displayed data is taken from the low 1024 x 768 or 640 x 480 bits in the buffer. Some non-visible display buffer storage is used for various housekeeping tasks. Some non-visible display storage is also available to PC applications for use as off-screen work areas.

More details are given in Appendix A, " Display Adapter 8514/A Compatibility."

Palettes

The XGA video output is by a palette (color look-up table). The palette allows a program to select the actual colors (or gray levels) that appear on the display from a greater range of colors than could be provided by mapping the number of bits per pel to fixed colors. The palette is a loadable direct access storage area that translates a particular pel value into three analog signals which drive the red, blue, and green guns of the display. The number of colors that can be displayed on the screen at any one time depends upon the number of bit planes which are enabled for display.

VGA Palette: Default palettes are provided so that a palette need not be loaded for every application. In VGA mode, the default palette is automatically selected, to conform with the display being driven, and loaded by the Basic Input/Output System (BIOS).

Advanced-Function Palette: In advanced-function mode, the default palette is optionally loaded by the adapter interface HOPEN order. Two default advanced-function palettes are provided, for color and monochrome displays respectively.

Appendix B, " Default Palettes," shows the palette diagram.

To use a VGA palette for an advanced-function application:

1. Switch to the VGA mode that uses the required palette.
2. Execute the HOPEN order with no default palette load selected.
3. Continue with adapter interface orders.

The default palettes for advanced-function mode are consistent with those for VGA modes. Some advanced functions (update mixes, for example), used to process images, or to interface with image or graphics data streams, require different palettes. Here are some examples:

- Four-bit (16-color) palette, based on the Color Graphics Adapter (CGA) colors.
- Four-plane monochrome palette, with a 16-level linear-step gray scale.
- Eight-plane (256-color) palette, (bits assigned: four green, two red, two blue) with linear intensity steps.
- Eight-plane monochrome palette, having a 64-level gray scale with linear steps (bit-compatible with four-plane monochrome).

When not in the advanced-function mode, any VGA palette-loading operation loads the same data into the advanced-function palette.

The advanced-function palette is used by both advanced-function and VGA modes to drive the display attached to the XGA video sub-system; it must be updated to suit the bit planes being displayed.

All-Points-Addressable (APA) Functions

XGA, with the XGA Adapter Interface, supports the following APA functions:

- Bit-block transfer (BITBLT)
- Line drawing
- Area filling
- Patterns
- Color mixing
- Scissoring
- Plot and step

- Vector read and write
- Red/green/blue masking.

For more details of these functions, see "Summary of Functions" on page 2-1.

Text and Alphanumeric Support

Two forms of character support are available:

Text Support

Text support is provided for applications that employ fonts of varying sizes on the screen, with "what you see is what you get" (WYSIWYG) viewing of text documents. Proportional character spacing is supported.

Character set definitions, which may be bit arrays or vector strings, are held in PC main storage.

The maximum character size that can be displayed is 255 x 255 pels.

Alphanumeric Support

Fixed-size character cells are used for basic alphanumeric operations such as listing directories and files. Character attributes are supported using fonts provided with the adapter.

The adapter supports IBM 3270 Information Display System alphanumeric operations through the XGA Adapter Interface. Blinking is not supported. Four character sets are provided as follows:

Cell size, pels	Characters per row	Rows per screen	Display size, pels
12 x 23	85	33	1024 x 768
12 x 20	85	38	1024 x 768
7 x 15	146	51	1024 x 768
8 x 14	80	34	640 x 480

Figure 1-1. Adapter Character Sets

When characters are written to the screen using this alphanumeric support, the alphanumeric data replaces any existing data in the cell positions written. If, therefore, an application requires that data overwritten in this way must be retained, the contents of the bit planes must be saved and restored.

Alternatively, the bit planes can be subdivided into two groups (or layers), for alphanumeric and APA operations respectively. (The default palette is designed to be used with two layers of four planes each.)

Fonts: The Adapter Interface installation program selects standard fonts as either "short stroke vectors" or "image." Five code pages are provided, in four sizes.

Code page details are given in Appendix C, " Font File Format."

More detail of Short Stroke Vector Characters is given under "Short Stroke Vector Characters" on page 3-72.

Code-page switching (CPS) support is part of the operating system function. No specific CPS support is included with the interface, although the default code page can be customized at installation time.

Cursors

A full-width cursor, which may be of any height, is provided for alphanumerics. Blinking is not supported.

Special adapter support is provided for graphics cursors. See "HSPRITE - Sprite at Given Position" on page 4-25 and "HSSPRITE - Set Sprite Shape" on page 4-26 for details of these commands.

Query Adapter or Display Configuration

PC programming can determine, by the Adapter Interface:

- Which display unit is attached, in terms of:
  - The pel density
  - Color or monochrome display.
- The current mode in which XGA is operating.

Byte and Bit Numbering

The adapter interface follows the Intel** and IBM PC data labelling convention:

- A "word" (WORD) is 16 bits
- A "doubleword" (DWORD) is 32 bits
- A "quadword" (QWORD) is 64 bits.

Byte Numbering Convention: Bytes are accessed using the IBM PC Ordering convention. The byte numbering convention is shown in the tables below.

Three integer formats are defined:

- Byte (BYTE) 8 bits
- Word (WORD) 16 bits
- Doubleword (DWORD) 32 bits.

Storage is addressed in units of a byte. The Least Significant Byte (LSB) of a word or doubleword resides in storage at the given address. The Most Significant Byte (MSB) of a word resides at an offset of +1 from the given address. The MSB of a doubleword resides at an offset of +3 from the given address.

Word

+1 MSB	+0 LSB

Doubleword

+3 MSB	+2	+1	+0 LSB

Figure 1-2. Byte Numbering Convention

** Trademark of the Intel Corporation. For a list of trademarks, see page iii.

Bit Numbering Convention: The bit numbering convention defined in the Base Function Set for doublewords, words, and bytes is shown in the figure below. The most significant bit (msb) has the largest bit number. The least significant bit (lsb) is at bit number 0.

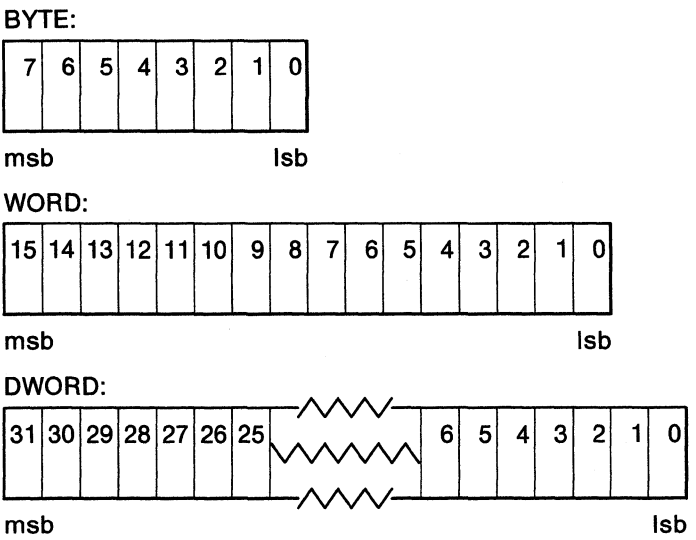

Figure 1-3. Bit Numbering Convention

Chapter 2. Adapter Interface Functions

The XGA Adapter Interface consists of a set of order entry points that can be called directly by the controlling environment. The interface driver code that incorporates these entry points is a serially reusable resource.

Summary of Functions

Adapter Interface functions are grouped in three sets; "Base Function Set," "Extended Function Set (1)" and "Extended Function Set (2)."

Base Function Set

The Adapter Interface Base Function Set includes the following graphic primitives:

- Lines
- Areas
- Markers/text
- Images
- Alphanumerics.

This set also includes the following graphic operations and drawing attributes:

- Color and mix control:
  - Color index
  - Color lookup table
  - Logic and arithmetic mix.
- Programmable primitives:
  - User-defined line types
  - User-defined area patterns
  - User-defined markers.
- Windowing assist:
  - Scissor
  - Restorable state.
- Layering assist:
  - Bit plane control
  - Color lookup table.

- Bit block transfer:
  - Between bit planes and memory
  - Between bit planes and bit planes.

Extended Function Set (1)

The Adapter Interface Extended Function Set (1) includes primitives for:

- Pel operations
- Read/write vector
- Bitmap operations.

Extended Function Set (2)

The Adapter Interface Extended Function Set (2) includes primitives for screen pointer support.

Adapter Interface Functional Model

Drawing Engine

The Adapter Interface functional model consists of a "drawing engine" which provides the function set to operate on the storage model.

Adapter Interface Concepts

The Adapter interface consists of a set of functions which define video memory models, together with a set of functions to manipulate data within the video memory models.

Bitplane and Bitmap Models

Video memory models can be visualized as either "bit planes" or "bitmaps."

For the bit plane model, the video memory configurations can be thought of as separate "planes" of data, one on top of the other. These planes are numbered 0 through 7. Plane 0 gives bit 0 of the color index, plane 1 gives bit 1 of the color index and so on. Typically, this model is used for Computer Aided Design type applications, where each plane forms an independent "layer" of the design.

The bitmap model regards video memory as a list of color indexes (pel values). Each pel can be represented by 1 through 8 bits, giving maximum pel values ranging from 1 to 255. This model is used for many applications, for example displaying images.

For either video memory model the final color of each screen pixel is formed by using the color index, to index into a palette table, containing red, green and blue intensity values. Individual planes can be enabled for update, display or both.

Note: The terms "bit planes," "APA storage," and "current bitmap" should be regarded as synonymous when they occur in this Technical Reference.

Lines

The adapter interface provides orders that draw polylines and relative lines from the current position, or another given position. The interface can also be used to draw disjoint lines.

Line type and line width can be programmed, as described at "HSLT - Set Line Type" on page 3-62 and "HSLW - Set Line Width" on page 3-64.

Each line type is defined by a list of pel counts "on" and "off." The following orders can be used in a list to form a continuous line, in which the start of the line pattern generated by an order matches the end of the line pattern generated by the previous order:

- HCLINE - Line at current position
- HCRLINE - Relative line at current position
- HSCP - Set current position.

All other line orders reset the line pattern count after each line.

Area Fill

Bounded Areas: Bounded areas are defined by a list of orders beginning with a Begin Area (HBAR) and ending with an End Area (HEAR). Between the Begin Area and End Area orders is a list of line drawing orders which define the boundaries of one or more closed areas.

The following orders are *not* valid between HBAR and HEAR:

- Control (HOPEN, HCLOSE, and HSMODE)
- Set Scissor (HSHS)
- Set color comparison (HSCMP)
- Set bit plane controls (HSBP)
- Character string orders (HCHST, HCCHST)
- Set area fill plane (HSAFP)
- Set bitmap (HSBMAP).

If one or more of these orders (for multiplane character sets) is used within an area definition, the effect of the area definition becomes undefined.

Note: Other drawing primitives (for example, *image*) which are allowed between HBAR and HEAR can change the "current position"; this effect has to be managed, as necessary, by the controlling environment.

Rectangular areas can usually be filled more rapidly by using the HRECT order instead of the HBAR, HEAR, and line orders.

Auxiliary APA Storage: Auxiliary APA storage has three uses in the adapter interface driver:

1. For drawing the boundary of an area to be filled
2. As a graphics text (image) cache
3. As a general work area for markers.

In general, the separate demands on this resource, although mutually exclusive, do not compete. However, if an area definition is suspended, any image graphics text drawn, invalidates the suspended definition. The controlling environment is responsible for managing this situation.

Note: The auxiliary APA storage can be used by an application to store bitmaps (see "HSBMAP - Set Bitmap Attributes" on page 4-18).

Image and Bit Block Transfer

The adapter interface functions support the following classes of image format:

- Across the planes (1 bit per pel)

  The source data is applied as a pattern. The foreground color is written, using the current mix, wherever there is a binary one in the pattern, and the background color and mix are used where there is a zero in the data.

- Through the planes (1,2,4, or 8 bits per pel)

  The source data is applied as a color index. The color indexes are written pel by pel, using the current foreground mix.

- Packed (1,2,4, or 8 bits per pel)

  If the bitmap copy order (HBMC) is given, a pattern bitmap can be employed to control the use of foreground and background mix. If no pattern is specified, the "0" color indexes are written using background mix and all other color indexes are written with the foreground mix.

The adapter interface supports the following bit block operations:

- Bit block output — write to bit planes
- Bit block input — read from bit planes
- Bit block copy — from bit plane(s) to bit plane(s)
- Bitmap copy — from bitmap to bitmap.

Bit block transfers are controlled by the following drawing processor attributes:

- Planes enabled for update
- Logic or arithmetic mix function
- Color (across the planes only)
- Scissor.

Control

The adapter interface supports a wide variety of drawing process control functions, including:

- Initializing the adapter interface (HOPEN, HINIT and HSYNC orders)
- Setting and querying modes (HSMODE and HQMODE orders)
- Limiting the area of update using the Scissor (HSHS order)
- Manipulating the color palette (HLDPAL, HSPAL and HRPAL orders).

For a complete list of the control orders with a detailed description, see "Control Orders" on page 3-28.

Attributes

For a definition of the adapter interface attributes associated with various drawing functions, such as color, mix and line type, see "Attribute Orders" on page 3-57.

Text Support

For a complete description of adapter interface text support, see "Programmable Character Set Specification" on page 3-70.

Alphanumerics Support

For details of adapter interface alphanumerics support, see "Alphanumeric Orders" on page 3-79.

Pel Operations

The function of pel operations is to read and write pel strings, and plot and step with variable source data.

Offscreen Bitmaps

An offscreen bitmap may be specified by using the HSBMAP order.

Many bitmaps may be defined. The current bitmap is the last one to be defined, and as such it is the "destination" for graphics update operations.

Using the Call Interface

A set of entry points, executable in the controlling environment, provides a call interface to the display adapter.

Issuing Orders

Orders are issued by calling order entry points. There is a separate entry point for each order. Before an entry point is called, the address of the parameter block must be put on the stack.

The Call Interface Specification

The call interface consists of a set of procedures for processing orders.

Entry Point Calling Procedure

Function: Each order or pseudo-order, (for example HEGS) has a procedure entry point. Not all orders have parameters, but all have parameter blocks.

Calling Sequence: An Adapter Interface call is invoked by an indirect call through a 32-bit pointer, passing a 32-bit pointer to a parameter block.

The input to adapter interface CALL is a pointer to an order parameter block, somewhere in the controlling environment.

DOS Environment Linkage Mechanism: In a DOS environment, the adapter interface microcode is a device driver, which is installed by including a DEVICE=name.SYS statement in the CONFIG.SYS file. A call table in the adapter interface driver module contains the 32-bit address of each entry point.

See Figure 2-2 on page 2-11 and Figure 2-3 on page 2-12.

Before calling the entry points, the controlling system must access the address of the adapter interface call table by the adapter linked list of structures (which is built up during installation) as follows:

```
MOV     AH,35H      ;
MOV     AL,7FH      ; Get interrupt vector 7FH ...
INT     21H         ; ... by function call to DOS.
MOV     AX,ES       ; OR vector 7FH segment ...
OR      AX,BX       ; ... with vector 7FH offset.
JZ      NOCODE      ; If NULL then interface code is not installed.
XOR     BX,BX       ;
MOV     AX,0113H    ; Get interface linked list address...
INT     7FH         ; ... by function call to the adapter.
JC      NOCODE      ; If CARRY then interface code is not installed.
                    ; CX:DX is the address of the linked list.
```

The linked list is then searched until the correct adapter type is found.

Compatibility: Any program, using the Interrupt 7F (0105) mechanism, (as supported by the 8514/A Display Adapter) which returns the address of a single call table in CX:DX, is fully supported, as follows:

```
MOV     AH,35H      ;
MOV     AL,7FH      ; Get interrupt vector 7FH ...
INT     21H         ; ... by function call to DOS.
MOV     AX,ES       ; OR vector 7FH segment ...
OR      AX,BX       ; ... with vector 7FH offset.
JZ      NOCODE      ; If NULL then interface code is not installed.
MOV     AX,0105H    ; Get interface call table address...
INT     7FH         ; ... by function call to the adapter.
JC      NOCODE      ; If CARRY then interface code is not installed.
                    ; CX:DX is the address of the call table.
```

Any 8514/A Display Adapter interface in the system, is included automatically in the linked list, to ensure that it can work in a multiple-adapter environment.

See Appendix A, " Display Adapter 8514/A Compatibility."

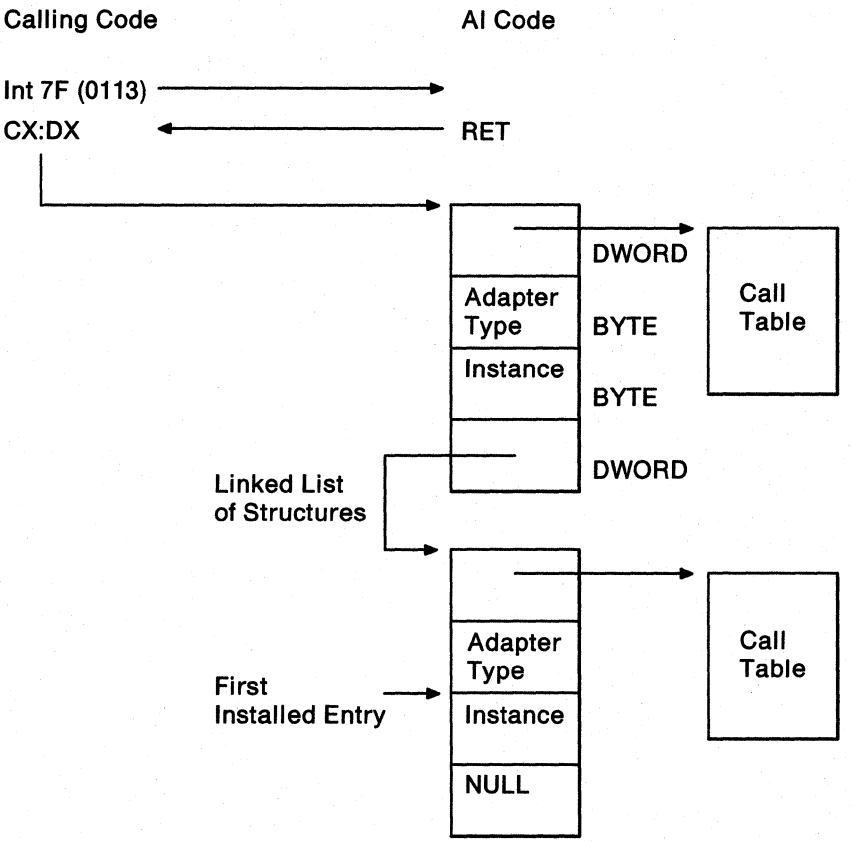

Figure 2-1. PC Environment Linkage Mechanism

The "adapter type" in the schematic (above) is identified by a two-digit code—see "HQMODE - Query Current Mode" on page 3-36.

"Instance" in the schematic is a number that indicates the occurrence of a particular type of adapter in the system (the first such instance being indicated by a 1, and so on).

An instance of an Adapter Interface Entry Point Table is shown on page 2-11.

Hex	Entry Point	Order
00	HLINE	Polyline
04	HCLINE	Polyline at current position
08	HRLINE	Relative polyline
0C	HCRLINE	Relative polyline at current position
10	HSCP	Set current position
14	HBAR	Begin area
18	HEAR	End area
1C	HSCOL	Set foreground color
20	HOPEN	Open adapter
24	HSMX	Set mix
28	HSBCOL	Set background color
2C	HSLT	Set line type
30	HSLW	Set line width
34	HEGS	Erase graphics screen
38	HSGQ	Set graphics quality
3C	HSCMP	Set color comparison register
40	HINT	Interrupt order
44	HSPATTO	Set pattern reference point
48	HSPATT	Set area fill pattern shape
4C	HLDPAL	Load palette
50	HSHS	Set hardware scissor
54	HBBW	Bit block write
58	HCBBW	Bit block write at current position
5C	HBBR	Bit block read
60	HBBCHN	Bit block chained data
64	HBBC	Bit block copy
68	HSCOORD	Set coordinate format
6C	HQCOORD	Query coordinate format
70	HSMODE	Set mode
74	HQMODE	Query current mode
78	HQMODES	Query possible modes
7C	HQDPS	Query drawing process state size
80	HRECT	Fill rectangle
84	HSBP	Set bit plane controls
88	HCLOSE	Close adapter
8C	HESC	Escape — terminate processing
90	HXLATE	Assign MP text color index
94	HSCS	Set character set
98	HCHST	Character string
9C	HCCHST	Character string at current position

Figure 2-2. Entry Point Table (1 of 2)

Hex	Entry Point	Order
A0	ABLOCKMFI	Write MFI character block
A4	ABLOCKCGA	Write CGA character block
A8	AERASE	Erase rectangle
AC	ASCROLL	Scroll rectangle
B0	ACURSOR	Set cursor position
B4	ASCUR	Set cursor shape
B8	ASFONT	Set character set
BC	AXLATE	Assign alpha color index
C0	HINIT	Initialize state
C4	HSYNC	Synchronize adapter
C8	HMRK	Marker
CC	HCMRK	Marker at current position
D0	HSMARK	Set marker shape
D4	HSLPC	Save line pattern count
D8	HRLPC	Restore line pattern count
DC	HQCP	Query current position
E0	HQDFPAL	Query default palette
E4	HSPAL	Save palette
E8	HRPAL	Restore palette
EC	HSAFP	Set area fill plane
F0	ASCELL	Set alpha character size
F4	ASGO	Set alpha grid origin
F8	HDLINE	Disjoint line
FC	Reserved	
100	HPEL	Write pel string
104	HRPEL	Read pel string
108	HPSTEP	Plot and step
10C	HCPSTEP	Plot and step at current position
110	HRSTEP	Read and step
114	HSBMAP	Set bitmap attributes
118	HQBMAP	Query bitmap attributes
11C	HBMC	Bitmap copy
120	HSDW	Set display window
124	HSPRITE	Sprite at given position
128	HSSPRITE	Set sprite shape
12C	HRWVEC	Read/write vector
130	Reserved	
134	Reserved	
138	HSFPAL	Save full palette
13C	HRFPAL	Restore full palette
140	HQDEVICE	Query device-specific information

Figure 2-3. Entry Point Table (2 of 2)

Calling Mechanism

DOS Environment: In the DOS environment, each entry point is called by:

1. Pushing the 32-bit address of the order parameter block onto the stack.
2. Calling the entry point with a CALL FAR.

In the DOS environment, the adapter interface call is returned with the RETURN FAR instruction, which removes the parameters used from the stack.

The following registers are preserved across the call:

- EBP, ESI, EDI, CS, DS, SP, SS.

All other registers may be changed.

State Data

The state data in the adapter interface is classified as either:

- Task-independent, or
- Task-dependent.

In a single-tasking environment like DOS, there is normally only one copy of a task-dependent state; it would be possible however, to define a number of different states and switch between them (to support multiple applications, for example). When an HOPEN order is processed the task-independent state is set to the adapter defaults. (The adapter initializes to the maximum screen size possible). At HINIT time, the task-dependent state is initialized to the default values.

The XGA Display Adapter/A defaults for both states are given in the following sections.

Task-Dependent State: The task-dependent state is that part of the state which varies between tasks:

- Current position: 0,0
- Current drawing attributes:
  - General:
    - Foreground color: white (with default palette loaded)
    - Background color: black (with default palette loaded)
    - Foreground mix: overpaint
    - Background mix: leave alone

- Comparison color: not initialized
- Comparison logic: false.

– Lines:
  - Line type: solid
  - User line definition: not initialized
  - Line width: 1
  - Line pattern position: not initialized
  - Saved line pattern position: not initialized
  - In area flag: reset.
– Areas:
  - Scanning rectangle: not initialized
  - Pattern definition: solid
  - Pattern origin: 0,0.
– Text:
  - Text control block pointer: not initialized.
– Markers:
  - Marker shape: not initialized.
– Bitmaps:
  - Current bitmap attributes: initialized.
– Current drawing controls:
  - Scissor rectangle: full screen
  - Graphics quality flags: line draw set to last pel null
  - Plane enable mask: set to all planes for update
  - Multiplane text color index translate table: eight linear values (0 to 7).
– Alphanumeric state data:
  - Cursor position: top left of screen
  - Cursor definition: invisible
  - Plane enable mask: set to all planes
  - Color index translate table:

    16 linear values (0 to 15) for foreground

    16 linear values (0 to 15) for background.
  - Character set definition: not initialized.

Task-Independent State: The task-independent state data is common to all tasks:

- Default color palette data set:
  - 16 default CGA color set (color monitor attached)
  - 16 optimized gray shade mapping of default color set (monochrome monitor attached).

The rest of the palette is loaded as follows:

16 x color 1
16 x color 2
.
.
.
16 x color 15

- Adapter mode is set by user at HOPEN time; it defaults to mode 0 (for 1024 x 768 resolution display) or mode 1 (for 640 x 480 resolution display)
- Task-dependent state buffer pointer: **not initialized**
- Plane display mask: set to all planes
- Working storage: **initialized**
- Screen pointer:

  Sprite position: 0,0
  Sprite shape: **not initialized**
  Sprite attribute: invisible.

State Control: Initially, the interface driver has to be given the task-independent state buffer at link time, when the buffer is initialized by an HOPEN order. The task-dependent buffer pointer is passed either by HINIT (for uninitialized buffers) or by HSYNC (for previously initialized buffers). The following operations can be achieved using the adapter interface control entry points:

- Initialize the adapter - HOPEN followed by an HINIT
- Initialize a new task - HINIT
- Perform a task switch - HSYNC
- Change state data - HSYNC
- Change mode - HSMODE
- Query mode - HQMODE
- Switch out of adapter interface mode - HCLOSE.

Data Formats

The data formats defined here are the "templates" used to locate data within the fields defined in the orders. An individual process might only use part of the data. For example, the base set drawing process ignores the fractional part of coordinates and those bits in the integer part that are not supported by a particular hardware device.

Absolute Coordinate Data: An absolute coordinate is a 2-byte, twos complement binary number.

In the order definitions, the length of an absolute coordinate is denoted by c, with a value of 2 for the implementation described in this book. The length of a pair of absolute coordinates is denoted by p, with a value of 4 for the implementation described in this book.

31 sign	30 msb	...	16 lsb	15 sign	14 msb	...	0 lsb
y				x			

Figure 2-4. Format of a Pair of Absolute Coordinates

- The origin of screen coordinate space is in the top left corner, from which an increasing value of x moves to the right, and an increasing value of y moves downwards.
- The coordinate origin (x = 0, y = 0) and incremental directions for the XGA Display Adapter/A are the same as those for the screen.
- The XGA Display Adapter/A coordinate range in both x and y axes is from −2048 to +6143. A line can be drawn in any part of this coordinate range, but the change it produces in x or y coordinates must be not greater than 4096.
- The screen coordinate ranges are:
  - x from 0 to 639 or 1023
  - y from 0 to 479 or 767.
- The XGA Display Adapter/A supports two-dimensional coordinates with no fractions and 1-byte relative offsets.

Relative Coordinate Data: A relative coordinate is a 1-byte, twos complement binary number.

The length of a pair of relative coordinates is denoted by r, with a value of 2 for the implementation described in this book.

15 sign	14 msb	...	8 lsb	7 sign	6 msb	...	0 lsb
ry				rx			

Figure 2-5. Format of a Pair of Relative Coordinates

Address Data: At the call interface, entry point parameters refer to areas of memory in the controlling environment. The call interface address data format depends on the controlling environment.

In the IBM PC AT or PS/2 environment, call interface address data is represented in doubleword (32-bit) fields.

The data format is "segment and offset" (Intel 80386).

For 80286 segmented addressing, the "word at displacement +2" is the segment selector of the data. The "word at displacement 0" is the offset address of the data. Within each word, the least significant byte has the lowest address in storage.

+3	+2	+1	+0
MSB	LSB	MSB	LSB
Segment Part		Offset Part	

Figure 2-6. Intel 80286 Address Data

The hardware address is generated using the Intel 80286 or 80386 address mapping mechanism. It is the responsibility of the controlling system to ensure that the adapter microcode has an appropriate level of access (read, write or execute) to the data.

Warning: Some LIM EMS emulators for the Intel 80386 do not implement Physical Address Services. Programmers are advised **not** to put adapter interface data objects in LIM EMS memory blocks.

Chapter 3. The Base Function Set

As described in Chapter 2, order definitions use the following symbols:

- **c** denotes the length (in bytes) of one absolute coordinate.
- **r** denotes the length (in bytes) of a pair of relative coordinates.
- **p** denotes the length (in bytes) of a pair of absolute coordinates.
- **s** denotes the length (in bytes) of a variable length string.

For the implementation described in this book, c=2, r=2, p=4, and s is variable.

Line Drawing Orders

Line Types

The adapter interface supports three types of line, defined by **vertex, offset,** and **disjoint** lists. There are two orders for vertex and offset lines: one for "given start position" and the other for "start at current position." There is only one order for disjoint lines.

Vertex Lines (HLINE and HCLINE): These lines are defined by lists (length 0—n) of coordinates.

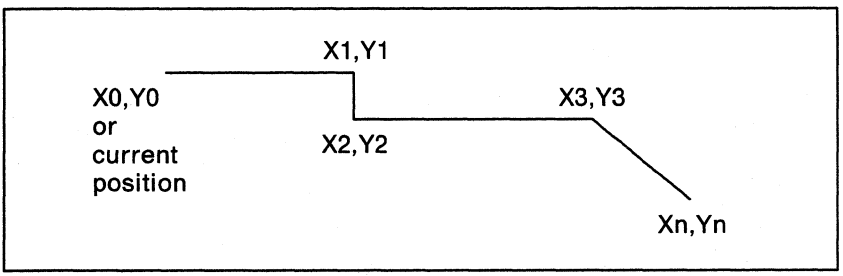

Figure 3-1. A Vertex Line List

Offset Lines (HRLINE and HCRLINE): An offset line list defines a line in terms of offsets from a previous vertex.

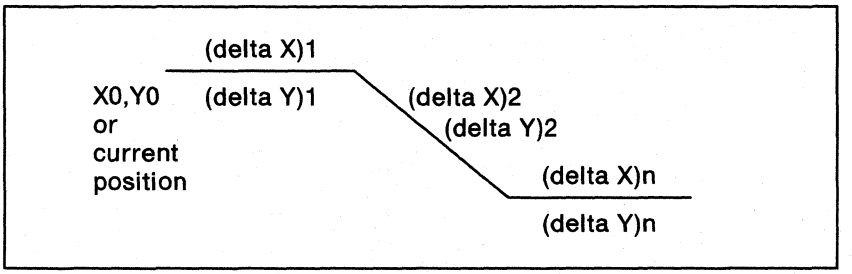

Figure 3-2. An Offset Line List

Disjoint Lines (HDLINE): A series of disjoint lines is defined by lists that give the start and end coordinates of each line.

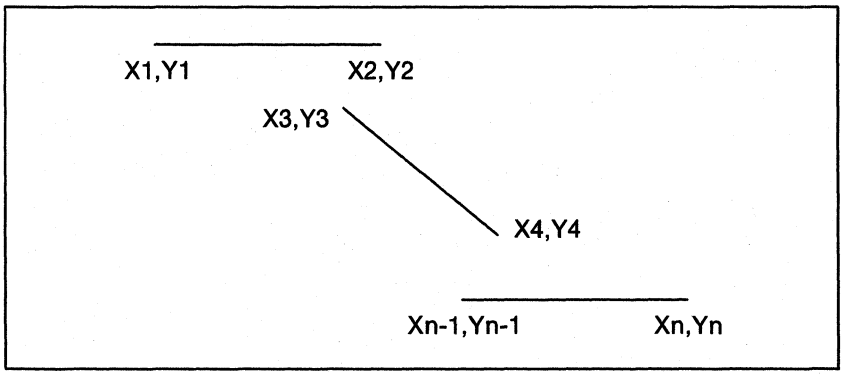

Figure 3-3. A Disjoint Line List

Note: Vertex and offset lists are part of the adapter interface Base Function Set, but disjoint lists are part of Extended Function Set 1, as described at "HDLINE - Disjoint Line" on page 4-2.

Line Order Attributes

The attributes of the line orders HLINE, HCLINE, HRLINE, HDLINE, and HCRLINE which may be specified are:

- Line type
- Line width
- Foreground and background color
- Color mix
- Color comparison.

Line Order Control

The line orders HLINE, HCLINE, HRLINE, and HCRLINE are subject to control by:

- "Last pel" processing, which affects the way in which lines are drawn.

  See "HSGQ - Set Graphics Quality" on page 3-40.

- The Scissor function as specified by the HSHS order.

  See "HSHS - Set Scissor" on page 3-42.

HLINE - Line at Given Position

Function: The HLINE order defines zero or more connected straight lines.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥p)
2	P0	Coordinate data of line start
2+p	P1	Coordinate data of first line end
.		
.		
.		
2+np	Pn	Coordinate data of n^{th} line end

Figure 3-4. HLINE: Line at Given Position

Description: A line is drawn from a point P0 to another point P1, then from P1 to P2, from P2 to ... Pn-1, and from Pn-1 to Pn. Any number of points can be present, bounded only by the maximum order length. Consecutive points in the order are joined by straight lines.

Usage Notes

- An HLINE with the coordinates for only one point specified, does not draw a line, but just sets the current position.
- Any number of points can be specified, bounded only by the maximum order length.
- HLINE resets the line pattern count, so that the next line starts at the beginning of a line pattern as described by the line type.
- Current position is set to the last point specified.

Attributes

- Line type
- Line width
- Foreground and background color
- Color mix
- Color comparison.

HCLINE - Line at Current Position

Function: The HCLINE order defines zero or more connected straight lines.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥0)
2	P1	Coordinate data of first line end
2+p	P2	Coordinate data of second line end
.		
.		
.		
2+(n-1)p	Pn	Coordinate data of n^{th} line end

Figure 3-5. HCLINE: Line at Current Position

Description: A line is drawn from the current position (CP) to another point P1, then from P1 to P2, from P2 to ... Pn-1, and from Pn-1 to Pn. Any number of points can be present, bounded only by the maximum order length. Consecutive points in the order are joined by straight lines.

Usage Notes

- Any number of points can be specified, bounded only by the maximum order length.
- HCLINE does not reset the line pattern count.
- Current position is reset to the last point specified.

Attributes

- Line type
- Line width
- Foreground and background color
- Color mix
- Color comparison.

HRLINE - Relative Line at Given Position

Function: The HRLINE order defines zero or more connected straight lines, as with HLINE, except that the end point of each line is given as an offset from the start of the line instead of absolute coordinates.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥p)
2	P0	Coordinate data of line start
2+p	OFF1	Offset data of first line end relative to the start point
2+p+r	OFF2	Offset data of second line end relative to the first line end
.		
.		
.		
2+p+(n-1)r	OFFn	Offset data of n^{th} line relative to $(n-1)^{th}$ line end

Figure 3-6. HRLINE: Relative Line at Given Position

Description: Line offsets are added cumulatively to a first point (P0) to generate a sequence of points (P1, P2, ..., Pn) where n is the number of offsets specified:

```
Pa = P0 + OFF1 + OFF2 + ... + OFFa
for 1 ≤ a ≤ n
```

Straight lines are drawn connecting the points in sequence, from P0 to Pn.

Usage Notes

- An HRLINE with no offsets specified, does not draw a line but just sets the current position.
- Any number of offsets can be included, bounded only by the maximum order length.
- HRLINE resets the line pattern count.
- Current position is set to the last point specified, or the line start position if no offsets have been specified.

Attributes

- Line type
- Line width
- Foreground and background color
- Color mix
- Color comparison.

HCRLINE - Relative Line at Current Position

Function: The HCRLINE order defines zero or more connected straight lines, as with HCLINE, except that the end point of each line is given as an offset from the start of the line, instead of absolute coordinates.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥0)
2	OFF1	Offset data of first line end relative to its start point
2+r	OFF2	Offset data of second line end relative to the first line end
.		
.		
.		
2+(n-1)r	OFFn	Offset data of n^{th} line end relative to the $(n-1)^{th}$ line end

Figure 3-7. HCRLINE: Relative Line at Current Position

Description: The offsets are added cumulatively to the current position (CP) to generate a sequence of points P1, P2, ..., Pn, where n is the number of offsets specified.

```
Pa = CP + OFF1 + OFF2 + ... + OFFa
for 1 ≤ a ≤ n
```

Straight lines are drawn connecting the points in sequence, from CP to Pn.

Usage Notes

- Any number of offsets can be included, bounded only by the maximum order length.
- HCRLINE does not reset the line pattern count.
- Current position is set to the last point specified.

Attributes

- Line type
- Line width
- Foreground and background color
- Color mix
- Color comparison.

Area Fill Orders

The "area fill" function allows an area to be specified by line and pel orders and then filled using various attributes including color and pattern. There is also a special order (HRECT) for filling rectangles. See "HRECT - Fill Rectangle" on page 3-13.

Line drawing orders that follow the HBAR order and precede an HEAR order, cause lines to be drawn in the area fill buffer in the format required to fill an area. The attributes used to fill the area are those current when the HEAR order is interpreted.

An "area definition" consists of boundary drawing orders between an HBAR order and an HEAR order. The area boundary consists of one or more closed figures. Each closed figure is made up of a contiguous set of line objects. The first closed figure within an area is defined as starting at the first drawing order after the HBAR order. This figure is delimited either by an HEAR order, or any "move" order that is valid within an area definition, such as HSCP, HLINE or HRLINE. Delimiting a figure by closing within an area definition is referred to as the "autoclose" function. Autoclose can be enabled or disabled by using the HSGQ order.

The figures formed in this way jointly define the area boundary. Any region connected to the filled area is shaded for that portion of its area which lies within the Scissor rectangle.

When an area is shaded, left top boundaries are treated as part of the area, and shaded. Right bottom boundaries are not treated as part of the area, and are unchanged by the area fill.

The HBAR and HEAR orders do not change the value of the current position. The value of the current position is, however, changed by those orders used to define the area boundary, including any implicit "figure-closing" orders that may have been required, for example HSCP, HLINE or HRLINE.

Area orders may not be "nested"; if they are, the effect of all area definitions involved becomes undefined.

HBAR - Begin Area

Function: The HBAR order is used to indicate the beginning of a set of orders that defines the boundary of an area. The end of this set is indicated by an End Area (HEAR) order.

Entry Point Parameter Block: The HBAR order has a "dummy" parameter block of length zero.

Description: The HBAR order sets the drawing process to area drawing mode.

HEAR - End Area

Function: The HEAR order is used to indicate the end of a set of orders that define the boundary of an area.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (1)
2	BYTE	Flags: Bits 7—6: = 00: Fill area previously defined = 01: Suspend area definition = 10: Terminate area definition Bits 5—0: Reserved

Figure 3-8. HEAR: End Area

Description: The HEAR order identifies the end of an area definition.

The HEAR order causes the area inside the special outline drawn in the area fill buffer to be shaded. The shaded area is then merged with the picture using the current values of pattern, color and mix.

Usage Notes

- The maximum area fill pattern size is 32 x 32 pels.
- If the "Fill area previously defined" option is selected, the area fill algorithm is applied within the area outline.
- If the "suspend area definition" option is selected, the area outline is left in the area fill buffer. However, when an area definition is resumed (using the HBAR order) processing continues on the area previously suspended. If any attribute values needed for the fill operation are corrupted during suspension, it is the responsibility of the controlling environment to restore them.
- If the "terminate area definition" option is selected, no area is filled and the boundary definition is cleared.

Attributes

- Pattern—as set by the HSPATT order
- Foreground and background color
- Color mix
- Color comparison.

Areas are subject to control by the Set Scissor function.

The HEAR order does not change the current position. The current position is, however, changed by the orders that define the area boundary.

HRECT - Fill Rectangle

Function: The HRECT order performs a rectangular fill at one or more positions.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data n(4+p)
2	COORD	Top left corner of first rectangle
2+p	WORD	Width of first rectangle
4+p	WORD	Height of first rectangle
6+p	COORD	Top left corner of second rectangle
6+2p	WORD	Width of second rectangle
8+2p	WORD	Height of second rectangle
.		
.		
.		
(4n-2)+(n-1)p	COORD	Top left corner of n^{th} rectangle
(4n-2)+np	WORD	Width of n^{th} rectangle
4n+np	WORD	Height of n^{th} rectangle

Figure 3-9. HRECT: Fill Rectangle

Description: The rectangular area specified in the parameter block is filled and then merged with the bit plane contents, using the current values of pattern, color, and mix. Like other drawing primitives, the HRECT update is limited to the part of the screen space that falls within the "hardware scissor" section.

Usage Notes: The HRECT order usually gives a faster fill than the HBAR, HEAR, and line orders. The multiple HRECT function is an extension to the IBM Display Adapter 8514/A Base Function Set; it is supported by Extended Function Set (1) adapters only.

Attributes

- Pattern, as set by the HSPATT order
- Foreground and background color
- Color mix
- Color comparison.

Areas defined by the HRECT order are subject to control by the Set Scissor function.

The current position is set to the top left corner of the final rectangle after it has been displayed.

HMRK - Marker at Given Position

Function: The HMRK order draws the current marker symbol at one or more positions.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥p)
2	P0	Coordinate data of first marker
2+p	P1	Coordinate data of second marker
.		
.		
.		
2+np	Pn	Coordinate data of n^{th} marker

Figure 3-10. HMRK: Marker at Given Position

Description: The HMRK order draws a marker symbol at one or more points in order space. The first marker is drawn at point P0. Further marker symbols are drawn at the remaining points specified in the order.

Usage Notes

- The current marker symbol is the marker drawn.
- The marker is positioned so that the top left corner of the marker cell is located at:

  Xpn-(cx/2), Ypn-(cy/2)

  that is, the positioning coordinate specifies the approximate center of the marker (where cx = marker width and cy = marker height).

Attributes

- Marker shape, as set by the HSMARK order
- Foreground and background color
- Color mix
- Color comparison.

Markers are subject to control by the Set Scissor function.

Current position is set to the last coordinate specified.

HCMRK - Marker at Current Position

Function: The HCMRK order draws the current marker symbol at one or more positions.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥0)
2	P1	Coordinate data of second marker
2+p	P2	Coordinate data of third marker
.		
.		
.		
2+(n-1)p	Pn	Coordinate data of n^{th} marker

Figure 3-11. HCMRK: Marker at Current Position

Description: The HCMRK order draws a marker symbol at one or more points. The first marker is drawn at the current position. Further marker symbols are drawn at the remaining points specified in the order.

Usage Notes:

- The marker is positioned so that the top left corner of the marker cell is located at:

  Xpn-(cx/2), Ypn-(cy/2)

  that is, the positioning coordinate specifies the approximate mid-point of the marker (where cx = marker width and cy = marker height).
- The current marker symbol is the marker drawn.

Attributes

- Marker shape, as set by the HSMARK order
- Foreground and background color
- Color mix
- Color comparison.

Markers are subject to control by the Set Scissor function.

The current position is set to the last coordinate specified. If no coordinate has been specified, the current position remains unchanged.

Image Orders

The adapter interface image orders allow data to be moved between program buffers and the current bitmap. The HBBW, HCBBW and HBBR orders define the format, shape and position of the data, and use a PC data buffer to store the image. The HBBCHN order defines the data buffer address and performs the data transfer. The HBBC order must be used to move the data within the current bitmap.

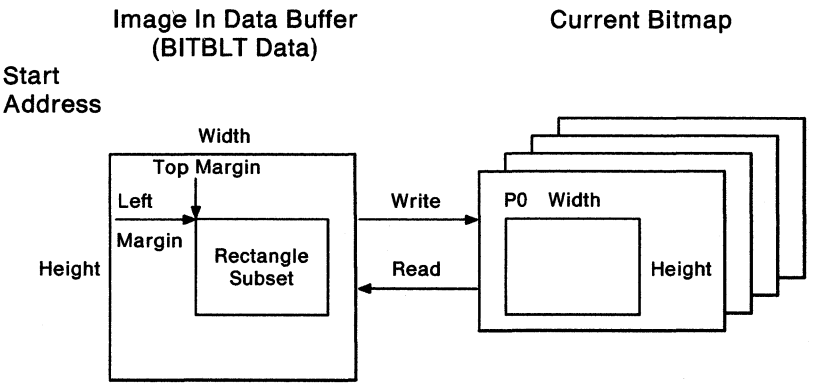

Figure 3-12. Adapter Interface Image Processing Schematic

For the "image in data buffer" (at left in the schematic), width and height are specified in bits for "across-the-plane" data, and in bytes for "through-the-planes" data.

For the "current bitmap" (at right in the schematic), if a subrectangle is specified, the bitmap width and height equal those of the subrectangle. Otherwise, bitmap width and height equal those of the full image.

Each one of the image orders HBBW, HCBBW, and HBBR specifies all the information given in the image processing schematic (above), except for the "start address" which is given in the HBBCHN order.

In the HBBC order, the current bitmap is both source and destination, and there is no rectangle subset definition.

HBBW - BITBLT Write Image Data

Function: The HBBW order identifies the start of a block of data that is to be written to the current bitmap.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥6+p)
2	WORD	The format of the BITBLT data: X'0000' "across the planes" X'0001' "packed" X'0008' "through the planes"
4	WORD	The width of the BITBLT data
6	WORD	The height of the BITBLT data
8	P0	Coordinates of the position at which the image data is drawn
8+p	WORD	Left margin (pels)
10+p	WORD	Top margin (pels)
12+p	WORD	Width of subrectangle
14+p	WORD	Height of subrectangle

Figure 3-13. HBBW: BITBLT Write Image Data

Description: The HBBW order defines a rectangular image at the given position. Subsequent HBBCHN orders pass data for the image. The size of the image block in the controlling environment is given by the BITBLT width and height parameters in pels (byte 4 to byte 7). If size is equal to 8+p bytes, then no subrectangle is defined.

The position of the image (top left corner) is given by P0 in pel coordinates.

If the order length field is equal to 14+p bytes, the left margin, top margin, width of subrectangle, and height of subrectangle parameters define a subrectangle contained within the PC data buffer image. When a subrectangle is specified, only the data within it is transferred by the BITBLT operation.

FORMAT X'0000': defines "across the planes" data. Each bit of data corresponds to pel. Wherever a binary 1 is encountered, the current color is written with the current mix. Wherever a binary 0 is encountered, the background color is written with the background mix.

For HBBR and HBBW BITBLT, the first pel in each row of data must begin on a byte boundary. Padding bits are ignored where the width within the BITBLT data is not exactly divisible by eight. If a subrectangle is defined,

then it is the full rectangle row of data which must begin on a byte boundary. There are no restrictions on the subrectangle row start or end positions.

FORMAT X'0001': specifies that the source data is "packed" and that the bit/pel values match the current bitmap. For example, if the current bitmap is 4 bits/pel, the format defines two pels per byte; if the current bitmap is 2 bits/pel, the format defines four pels per byte, and so on. This is an extension to the IBM Display Adapter 8514/A Base Function Set; it is supported by Extended Function Set (1) adapters only.

FORMAT X'0008': defines "through the planes" data in terms of 1 byte per pel. Each byte of data corresponds to a part of the order space picture element color index. Bytes are padded with zeros to the left to form the color index. The resulting color index is written to the bit planes under the control of the current mix. The planes are then enabled for updating.

Usage Notes: The image data is supplied by the HBBCHN order which follows it. The image is terminated by any image order other than HBBCHN. Each HBBCHN order supplies an arbitrary length of data. The image data may be split between one or more HBBCHN orders.

The image data is drawn as a "raster," row by row in adjacent pels, starting with the pel at the top left corner of the block defined by P0, and continuing along the first row of pels towards the top right corner. The second row begins with the left side pel immediately below the start of the first row, and continues towards the right.

When a subrectangle is defined, the length of data that should be chained is the size of the full rectangle, not the size of the subrectangle.

Color expansion is supported for data with bit/pel values from 1 bit per pel "across the planes" up to 8 bits per pel "through the planes."

Attributes

- Foreground and background color (for "across the planes" format only)
- Color mix
- Color comparison.

The image is subject to control by the Set Scissor function.

The current position is set to P0.

HCBBW - BITBLT Write Image Data at Current Position

Function: The HCBBW order identifies the start of a block of data that is to be written to the current bitmap.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥6)
2	WORD	The format of the BITBLT data: X'0000' "across the planes" X'0001' "packed" (bitmap) X'0008' "through the planes" (byte)
4	WORD	The width of the BITBLT data
6	WORD	The height of the BITBLT data
8	WORD	Left margin (pels)
10	WORD	Top margin (pels)
12	WORD	Width of subrectangle
14	WORD	Height of subrectangle

Figure 3-14. HCBBW: BITBLT Write Image Data at Current Position

Description: The HCBBW order defines an image block at the current position. Subsequent HBBCHN orders pass data for the image.

The size of the image block in the controlling environment is given by the width and height parameters in pels, (byte 4 to byte 7).

The position of the image (top left corner) is the current position.

If the order length field is equal to 14 bytes, the left margin, top margin, width of subrectangle, and height of subrectangle parameters define a subrectangle contained within the PC data buffer image. When a subrectangle is specified, only the data within it is transferred by the BITBLT operation.

FORMAT X'0000': defines "across the planes" data. Each bit of data corresponds to one pel. Wherever a binary 1 is encountered, the current color is drawn with the current mix. Wherever a binary 0 is encountered, the background color is drawn with the background mix.

The first pel in each row of PC data must begin on a byte boundary. Padding bits are ignored where the width is not exactly divisible by eight. If a subrectangle is defined, then it is the full rectangle row of data which must

begin on a byte boundary. There are no restrictions on the subrectangle row start or end positions.

FORMAT X'0001': specifies that the source data is "packed" and that the bit/pel values match the current bitmap. If, for example, the current bitmap is 4 bits/pel, the format defines two pels per byte; if the current bitmap is 2 bits/pel, the format defines four pels per byte, and so on. This is an extension to the IBM Display Adapter 8514/A Base Function Set; it is supported by Extended Function Set (1) adapters only.

FORMAT X'0008': defines "through the planes" data in terms of 1 byte per pel. Each byte of data corresponds to a part of an order space picture element color index. Bytes are padded with zeros to the left to form the color index. The resulting color index is written to the bit planes under the control of the current mix. The planes are then enabled for updating.

Usage Notes: The image data is supplied by the HBBCHN order which follows it. The image is terminated by any image order other than HBBCHN.

Each HBBCHN order supplies an arbitrary length of data. The image data may be split between one or more HBBCHN orders.

The image data is drawn as a "raster," row by row in adjacent pels, starting with the pel at the top left corner of the block defined by the current position and continuing horizontally along the first row of pels towards the top right corner. The second row begins with the left side pel immediately below the start of the first row and continues towards the right.

When a subrectangle is defined, the length of data that should be chained is the size of the full rectangle, not the size of the subrectangle.

Color expansion is supported for data with bit/pel values from 1 bit per pel "across the planes" up to 8 bits per pel "through the planes."

Attributes

- Foreground and background color ("across the planes" format only)
- Color mix
- Color comparison.

The image is subject to control by the Set Scissor function.

The current position is unchanged.

HBBR - BITBLT Read Image Data

Function: The HBBR order identifies the start of a block of data that is to be copied from the current bitmap.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥8+p)
2	WORD	The format of the BITBLT data: X'0000' "across the planes" X'0001' "packed" X'0008' "through the planes"
4	WORD	The width of the BITBLT data
6	WORD	The height of the BITBLT data
8	BYTE	"Across the planes" source bit plane
9	BYTE	Reserved
10	P0	Coordinates of the position from which the image data is read
10+p	WORD	Left margin (pels)
12+p	WORD	Top margin (pels)
14+p	WORD	Width of rectangle
16+p	WORD	Height of rectangle

Figure 3-15. HBBR: BITBLT Read Image Data

Description: The HBBR order defines a rectangular image at the given position. Subsequent HBBCHN orders receive data from the image.

The size of the image block is given by the WIDTH and HEIGHT parameters in pels (byte 4 to byte 7).

The position of the image (top left corner) is given by P0 in pel coordinates.

If the order length field is equal to 16 + p bytes, a subset of the stored image data is defined as a rectangle. The position and dimensions of this rectangle are specified (in pels) in terms of its left and top margins, width, and height. Data is read only into this rectangle subset. The rectangle in storage need not be byte-aligned. Data outside the rectangle is not corrupted by the read operation.

FORMAT X'0000': defines "across the planes" data. The data is extracted from the specified bit plane. Each bit of data corresponds to one order space picture element. The first pel in each row of data begins on a byte boundary. Padding bits (zeros) are added where the width is not exactly divisible by eight.

If a subrectangle is defined, then it is the full rectangle row of data which must begin on a byte boundary. There are no restrictions on the subrectangle row start or end positions.

FORMAT X'0001': specifies that the data is "packed" and that the bit/pel values match those in the current bitmap. If, for example, the current bitmap value is 4 bits/pel, the format defines two pels per byte; if the destination bitmap value is 2 bits/pel, the format defines four pels per byte, and so on. This is an extension to the IBM Display Adapter 8514/A Base Function Set; it is supported by Extended Function Set (1) adapters only.

FORMAT X'0008': defines "through the planes" data in terms of 1 byte per pel. Each byte of data corresponds to part of the color index of an order space picture element. The least significant byte of the color index is extracted and stored in the HBBCHN order.

The current position is set to P0.

Usage Notes

- The data for the image is supplied by the HBBCHN order. The image is terminated by any image order other than HBBCHN.
- Each HBBCHN order supplies an arbitrary buffer length. The image data may be split between one or more HBBCHN orders. If the chain is terminated before the image area has been completely read, the remainder of the image block is not read. If the chain provides an excessive area of storage, the unused storage is not modified.
- The image data is read row by row in adjacent pels, starting at the top left corner of the block defined by P0, and continuing along the first row of pels towards the top right corner. The second row begins with the left side pel immediately below the start of the first row, and continues towards the right.
- When a subrectangle is defined, the length of data that should be chained is the size of the full rectangle, not the size of the subrectangle.

HBBCHN - BITBLT Chained Data

Function: The HBBCHN order holds image data for the BITBLT orders HBBR and HBBW.

Entry Point Parameter Block

Byte	Content	Meaning
0	Word	Length of following data (6)
2	DWORD	Address of data in controlling system
6	WORD[1]	Length of data in controlling system

Figure 3-16. HBBCHN: BITBLT Chained Data

Description: The HBBCHN order defines data that may form all or part of an image block.

The data buffer is defined by an address and length given in the order.

The content of the data buffer supplied by the HBBCHN order is defined by the parameters of the HBBW, HCBBW, or HBBR order that precedes it.

The data addressed by an HBBCHN order must constitute a complete rectangle within the image.

The HBBCHN order does not change the current position.

BITBLT Operations: For a description of BITBLT operations, see "HBBW - BITBLT Write Image Data" on page 3-18, "HCBBW - BITBLT Write Image Data at Current Position" on page 3-20, and "HBBR - BITBLT Read Image Data" on page 3-22.

Usage Note: Length of data in the controlling system is in the range from 1 to X'FFFF'.

---

1 This field is DWORD in the "flat" model environment (Intel 80386).

HBBC - BITBLT Copy

Function: The HBBC order copies a BITBLT block within the bit planes.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (8+2p)
2	WORD	The format of the BITBLT data: X'0000' "across the planes" copy X'0001' "through the planes" copy
4	WORD	The width of the BITBLT data
6	WORD	The height of the BITBLT data
8	BYTE	"Across the planes" source bit plane
9	BYTE	Reserved
10	P0	Coordinates of the source data
10+p	P1	Coordinates of the destination

Figure 3-17. HBBC: BITBLT Copy

Description: The HBBC order copies a rectangular image from a given source position in order space to a given destination position.

If the two image areas overlap, the destination overlays the source. If the image areas do not overlap, the source is unchanged. The effect of copying a rectangle to an overlapping rectangle is as if the source data is stored in a separate buffer.

The size of the image block is given by the width and height parameters, in pels.

The source and destination positions refer to the top left corner of the image in pels.

FORMAT X'0000': defines "across the planes" data. The data is extracted from the specified bit plane. Wherever a binary 1 is encountered in the source bit plane, the current color is written with the current mix. Wherever a binary 0 is encountered, the background color is written with the background mix.

FORMAT X'0001': defines "through the planes" data. The color index for each picture element of the source image is written to the destination image using the current color mix.

Attributes

- Foreground and background color (for "across the planes" format only)
- Color mix
- Color comparison.

The image copy is subject to control by the Set Scissor function.

The current position is set to P1.

HSCP - Set Current Position

Function: The HSCP order sets the current position.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (p)
2	P0	Coordinate data

Figure 3-18. HSCP: Set Current Position

Description:

Current Position is set to the value specified in the HSCP order.

The HSCP order does not reset the line pattern.

Control Orders

HOPEN - Open Adapter

Function: The HOPEN order initializes the adapter interface.

Entry Point Parameter Block

Byte	Content	Meaning
0	Word	Length of following data (3)
2	BYTE	Flags: Bit 7 = 1 Do not clear graphics screen Bit 6 = 1 Do not load a default palette Bits 5—0: Reserved (must be 0)
3	BYTE	Mode
4	BYTE	Return flags: Bit 7 = 1 Driver hardware mismatch Bit 6 = 1 Control program reject Bit 5—0: Reserved

Figure 3-19. HOPEN: Open Adapter

Description: The HOPEN order initializes the task-independent state; this process includes loading a default palette. For details of the task-independent state, see "Task-Independent State" on page 2-15.

Usage Notes

- If byte 2 of the parameter block is 0, the adapter is initialized and the default palette is loaded. See "State Data" on page 2-13 for details of the initialization process.
- For details of byte 3 (mode) see "HSMODE - Set Mode" on page 3-35.
- If HOPEN returns 0 in byte 4, the "open" is successful.
- Bit 6 of byte 4 (control program reject) is for the use of control programs.
- The HOPEN order is **not** valid between an HBAR and an HEAR order.

See "State Data" on page 2-13 for details of states and tasks.

HCLOSE - Close Adapter

Function: The HCLOSE order switches the adapter out of adapter interface mode.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (1)
2	BYTE	Reserved (must be 0)

Figure 3-20. HCLOSE: Close Adapter

Description: The HCLOSE order switches the adapter to the system default mode.

Usage Notes

- The HCLOSE order is **not** valid between an HBAR and an HEAR order.
- HCLOSE clears the display RAM.

HQCP - Query Current Position

Function: The HQCP order returns the current position coordinates.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (p)
2	P0	Coordinate data

Figure 3-21. HQCP: Query Current Position

Description: The coordinates of the current position are returned in the data area provided.

HQDFPAL - Query Default Palette

Function: The HQDFPAL order returns the first 16 color index values.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (64)
2	DWORD	First palette entry
6	DWORD	Second palette entry
.		
.		
.		
62	DWORD	Sixteenth color index value

Figure 3-22. HQDFPAL: Query Default Palette

Description: The adapter returns the palette entries that give the following colors when the default palette is loaded:

Value	Color	Value	Color
0	Black	8	Gray
1	Blue	9	Light blue
2	Green	10	Light green
3	Cyan	11	Light cyan
4	Red	12	Light red
5	Magenta	13	Light magenta
6	Brown	14	Yellow
7	White	15	High intensity white

If the adapter does not support a color in its default palette then it responds with the palette entry:

X'FFFFFFFF'

Usage Notes

- Each color index value is 4 bytes.
- If the HQDFPAL order is executed before an HOPEN is performed, and the adapter interface code has already been installed, all the fields returned are valid.

HINIT - Initialize State

Function: The HINIT order sets the task-dependent data to an initial state.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (2)
2	WORD	Address of task state buffer

Figure 3-23. HINIT: Initialize State

Description: The HINIT order sets the task-dependent data to an initial state.

Usage Notes

- In the DOS environment, the address parameter is a segment only.
- See "State Data" on page 2-13 for a discussion of the task-dependent state.

HSYNC - Synchronize Adapter

Function: The HSYNC order synchronizes the adapter hardware with a given task state.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (2)
2	WORD	Address of task state buffer

Figure 3-24. HSYNC: Synchronize Adapter

Description: The HSYNC order sets the adapter state to match the task-dependent state that is supplied as a parameter.

Usage Notes

- HSYNC must not be used to switch between different hardware instances of the code: this produces indeterminate results.
- HSYNC should not be used to synchronize the hardware state of one adapter with that of another adapter.
- In the DOS environment, the address parameter is a segment only.

See "State Data" on page 2-13 for a discussion of the task-dependent state.

HINT - Interrupt

Function: The HINT order synchronizes with a hardware event or interrupt.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (4)
2	DWORD	Interrupt or event identifier

Figure 3-25. HINT: Interrupt

Description: This order returns control to the controlling environment when the requested event or interrupt occurs.

Usage Notes

- The controlling system can use this order to synchronize with a hardware event.
- Bits in the interrupt and event identifier are defined as follows:

  Bit 31: Frame flyback

  Bit 30: Hardware not-busy

  Bits 29—0: Reserved.
- The controlling system should have bits set in the interrupt/event identifier to specify the events that cause the order to complete. Following any of the specified events, the adapter clears all the "event" bits except the one that indicates the cause of the interrupt.

HSMODE - Set Mode

Function: The HSMODE order sets the adapter mode.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (1 or 2)
2	BYTE	Mode byte (cell and screen sizes in pels):

Mode	*Alpha cells*	*Cell size*	*Screen size*
00	85 x 38	12 x 20	1024 x 768
01	80 x 34	8 x 14	640 x 480
02	128 x 54	8 x 14	1024 x 768
03	146 x 51	7 x 15	1024 x 768

3	BYTE	Flags: Bit 7: = 0: Clear all display RAM = 1: Do not clear display RAM Bits 6—0: Reserved (must be 0)

Figure 3-26. HSMODE: Set Mode

Description: The adapter is set to the mode specified by the mode byte.

Usage Notes

- The HSMODE order:
  - Resets the Scissor to the screen size of the new mode
  - Resets the display and update masks to default values
  - Sets the current position to 0,0
  - Sets the cursor to "invisible"
  - Resets to the screen bitmap
  - Sets the number of bits per pel, up to the maximum supported by the mode.
- A mode of X'FF' is a reserved value.
- The HSMODE order is **not** valid between an HBAR and an HEAR order.
- The HSMODE order conditionally clears the display RAM.

HQMODE - Query Current Mode

Function: The HQMODE order returns data that specifies adapter mode and configuration.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥18)
2	BYTE	Mode number
3	WORD	Driver code level
5	BYTE	Adapter type
6	BYTE	Display type (reserved)
7	BYTE	Alpha cell width (pels)
8	BYTE	Alpha cell height (pels)
9	BYTE	Number of bit planes
10	WORD	Screen width (pels)
12	WORD	Screen height (pels)
14	WORD	Pels/inch horizontal
16	WORD	Pels/inch vertical
18	BYTE	Monochrome or color: =X'00': Monochrome =X'FF': Color
19	BYTE	Intensity levels[2]
20	BYTE	Software area fill plane required (= X'01')
21	BYTE	VGA Mode: = X'01': Supported = X'00': Not supported

Figure 3-27. HQMODE: Query Mode

Description: The controlling environment can use the HQMODE order to query the current adapter mode and configuration. This data is returned in the data area supplied in the parameter block. If the data length is greater than 18, then Bytes 19, 20, and 21 are returned also.

---

2 Intensity levels are specified as the number of significant bits in each palette color entry (that is, the number of bits in the digital-to-analog converter (DAC) for each color). Zero indicates a non-loadable palette.

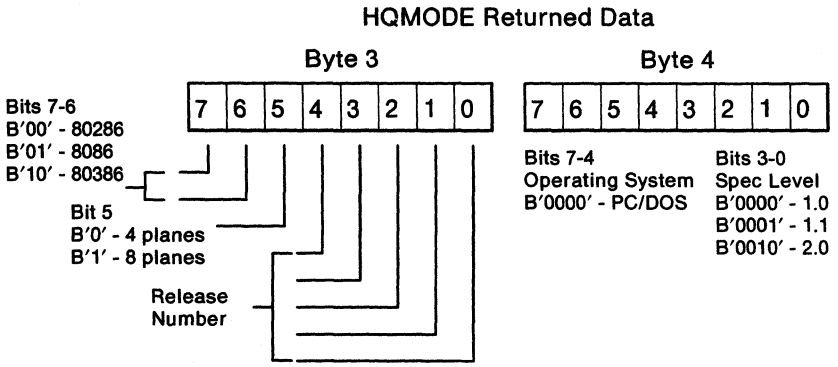

Figure 3-28. Code Level Numbering Convention

Usage Notes

- Each adapter type has been allocated a two-digit identifier which is "04" for this adapter.
- "Monochrome or color" and "intensity level" information enables users to load the palette correctly.
- Mode number: if the adapter is not "open," the message X'FF' is returned.
- If the HQMODE order is executed before an HOPEN is performed, and if the adapter interface code has been installed, the following fields returned are valid:
  - Code level
  - Adapter type
  - Number of bitplanes
  - Monochrome or color
  - Intensity levels.
- All other fields are invalid before an HOPEN command.
- The number of bit planes (byte 9) is the number of bits per pel in the current screen bitmap.

HQMODES - Query Adapter Modes

Function: The HQMODES order returns data specifying the modes that are available at the interface.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (33)
2	BYTE	Adapter type
3	DATA	Modes

Figure 3-29. HQMODES: Query Modes

Description: The controlling environment uses this order to identify the modes that are available. The mode data is returned in the data area supplied.

For a description of modes, see "HSMODE - Set Mode" on page 3-35.

Usage Note: Because the number of modes supported is variable, the returned data is terminated by an X'FF' byte. The controlling environment should have sufficient buffer space reserved for the "modes" data, (that is, 35 bytes).

HEGS - Erase Graphics Screen

Function: The HEGS order causes the screen to be cleared.

Description: The HEGS order has a dummy parameter block, and is used to clear the screen by setting the bit planes to zero. Attributes do not affect the screen clearing process.

The order sets the current position to the top left corner of the screen.

The area set to zero is limited by the "hardware scissor" function.

Planes that are not enabled for update are not modified by the HEGS order.

Usage Notes

- Because the HEGS order has a dummy parameter block, the length field must be zero.
- The order has the effect of clearing the current bitmap.

HSGQ - Set Graphics Quality

Function: The HSGQ order is used to set miscellaneous drawing attributes.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (2)
2	WORD	Flag settings: Bits 15—13 = B'0': Reserved Bits 12—11: = B'00': Last pel null on = B'01': Last pel null off = B'10': Conditional last pel Bit 10: = B'0': Autoclose on = B'1': Autoclose off Bits 9—0: Reserved (must be 0)

Figure 3-30. HSGQ: Set Graphics Quality

Usage Notes:

- Last pel null.

  When this bit is set, vector drawing always draws the last pel. If it is not set, the last pel is not drawn. This is pattern dependent; if the pattern specifies that a pel should be drawn, the above criteria apply.

- "Conditional last pel," means lines drawn in overpaint/AND/OR mixes, are drawn including first and last pel. Polylines drawn in other mixes (for example, XOR) are drawn with an adapter-dependent approximation.

- Conditional last pel processing, when the foreground mix is one of the following, is:
  - X'04' XOR
  - X'08' add
  - X'09' subtract (screen-new)
  - X'0A' subtract (new-screen)
  - X'0B' average
  - X'12' not screen and new
  - X'16' screen XOR new
  - X'18' not screen and not new
  - X'19' not screen XOR new
  - X'1A' not screen
  - X'1B' not screen or new
  - X'1E' not screen or not new.

  Then each line segment within the line orders (such as HLINE) is drawn with the last pel null. Otherwise the last pel of each line segment is drawn.
- Autoclose on or off; if autoclose is enabled, then any "move" type of order encountered during a boundary definition forces a closure from the current position to start position. See "HBAR - Begin Area" on page 3-10.

HSHS - Set Scissor

Function: The HSHS order causes the drawing process Scissor rectangle to be set, and optionally sets a Z Buffer map for non-rectangular scissoring.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (0 or 8 or 13)
2	WORD	Left limit of the rectangle (0)
4	WORD	Right limit of the rectangle (1)
6	WORD	Bottom limit of the rectangle (2)
8	WORD	Top limit of the rectangle (3)
10	ADDRESS	Pointer to Z Buffer map
14	BYTE	Flag Settings:
		Bits 7—0: Reserved (must be X'80')

Figure 3-31. HSHS: Set Scissor

Description: The scissor rectangle is set to the values specified in the HSHS order.

If the data length is 0, the Scissor function is disabled.

If the order length is equal to 13, then a Z Buffer map is defined.

If the left limit is to the right of the right limit, or the bottom limit is above the top limit, all graphic objects are discarded.

A 1 bit per pel Z Buffer is supported, if supplied. The content of the buffer defines the depth of the update data relative to the existing data. A 1 defines the update data as being in front of existing data, and hence the update data is drawn. A 0 defines the update data as being behind the existing data, and hence it is not drawn. The Z Buffer can be up to screen width and height. The width, height, and origin are specified by the limit coordinates, that is, width of the Z Buffer is right limit minus left limit, and height is bottom limit minus top limit.

All drawing operations are limited to the rectangle defined by the Scissor.

Usage Notes

- The non-rectangular Scissor function, specified by the pointer to Z Buffer map operand, is an extension to the IBM Display Adapter 8514/A Base Function Set; it is supported by Extended Function Set (2) adapters only.

- The drawing process Scissor inhibits the drawing of objects outside the rectangle defined in screen coordinate space, or inside the rectangle if on a 0 location (if Z Buffer map supplied). The boundary of the rectangle is inclusive, that is, pels on the boundary are drawn.
- The scissoring is performed by the drawing process after vector-to-raster conversion.
- Objects that fall outside screen space may be drawn within the scissor window. In this case an additional clip function is required to limit the data to adapter coordinate space.
- This order is not valid between an HBAR and an HEAR order.

HLDPAL - Load Palette

Function: The HLDPAL order loads a palette into the color lookup tables.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥1)
2	BYTE	Palette identification: Bits 7—1: Reserved Bit 0: 0: Load user-defined palette 1: Load default palette
3	BYTE	Reserved
4	WORD	Number of first entry to be loaded
6	WORD	Number of entries to be loaded
8	DWORD	Address of the palette entries in storage

Figure 3-32. HLDPAL: Load Palette

Description: The HLDPAL order loads the color lookup table with a pre-defined or user-defined palette. The address of the palette entries is a program (virtual) address in the controlling environment.

Each palette entry is a 4-byte field. The first byte defines the red intensity, the second byte defines the blue intensity, and the third byte defines the green intensity. The fourth byte is reserved. Each intensity definition is an integer between 0 and 255; 255 represents the maximum intensity.

The HLDPAL order allows a part of the color lookup table to be loaded. All other parts of the color lookup table are left unmodified.

Usage Note: If the default palette is specified, bytes 3—8 are redundant and need not be supplied.

HSPAL - Save Palette

Function: The HSPAL order saves the contents of the color palette and the display mask.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (defined by HQDPS)
2	DATA	Buffer

Figure 3-33. HSPAL: Save Palette

Description: This order saves the contents of the palette and the display mask, that can later be restored using the HRPAL order.

Usage Notes

- The HSPAL order, together with the HRPAL order, can be used to preserve palette integrity during operations which may corrupt it.
- The buffer size necessary for the palette contents is returned as a parameter of the HQDPS order.
- The color palette has 256 entries; the high-order bits of any color index beyond this range are ignored.
- Only the most significant 6 bits of each byte in the palette entries are used.
- This order is valid before an HOPEN has been performed.

HRPAL - Restore Palette

Function: The HRPAL order restores the contents of the color palette and the display mask.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (defined by HQDPS)
2	DATA	Buffer

Figure 3-34. HRPAL: Restore Palette

Description: This order restores the contents of the palette and the display mask, as saved previously by the HSPAL order.

Usage Notes

- The HRPAL order, together with the HSPAL order, can be used to preserve palette integrity during operations that may corrupt it.
- The buffer size necessary for the palette contents is returned as a parameter of the HQDPS order.
- The HRPAL order is valid before an HOPEN has been performed.

HSLPC - Save Line Pattern Count

Function: The HSLPC order saves the current line pattern count.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (0 or 2)
2	WORD	Area in which line pattern count is saved

Figure 3-35. HSLPC - Save Line Pattern Count

Description: The HSLPC order saves the current line pattern count, that can later be restored using the HRLPC order.

Usage Notes

- The HSLPC order, together with the HRLPC order, can be used to achieve continuity in a patterned line which straddles a scissor boundary.
- If the length field is 0, the line pattern count is saved in an internal buffer; if the length field is 2, the count is saved in bytes 2 and 3 of the parameter block.
- The ability to save the line pattern count to the parameter block, is an extension to the IBM Display Adapter 8514/A Base Function Set; it is supported by Extended Function Set (1) adapters only.

HRLPC - Restore Line Pattern Count

Function: The HRLPC order restores the saved line pattern count.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (0 or 2)
2	WORD	Area in which line pattern count is saved

Figure 3-36. HRLPC - Restore Line Pattern Count

Description: The HRLPC order restores a line pattern count that has been saved using the HSLPC order.

Usage Notes

- The HRLPC order, together with the HSLPC order, can be used to achieve continuity in a patterned line which straddles a scissor boundary.
- The value that is restored, is either the adapter default (if no HSLPC orders have been issued since initialization) or the value saved by the last HSLPC order.
- If the length field is 0, the line pattern count is restored from an internal buffer; if the length field is 2, the count is restored from the supplied field.
- The line pattern count is ≥ 0 and $<$pattern size.
- The ability to save the line pattern count to the parameter block, is an extension to the IBM Display Adapter 8514/A Base Function Set; it is supported by Extended Function Set (1) adapters only.

HSBP - Set Bit Plane Controls

Function: The HSBP order selects bit planes as required for updating and display, and controls the use of the palette.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (12 or 26)
2	DWORD	Planes selected for update bit mask (graphics or text)
6	DWORD	Planes selected for update bit mask (alphanumerics)
10	DWORD	Planes enabled for display bit mask
14	BYTE	Flags: Bits 7—0: Reserved(must be 0)
15	BYTE	Reserved
16	DWORD	Green bits mask
20	DWORD	Red bits mask
24	DWORD	Blue bits mask

Figure 3-37. HSBP: Set Bit Plane Controls

Description: The plane selected for the update bit mask specifies the planes to be updated under control of the drawing mix (see "HSMX - Set Mix" on page 3-67). All other planes remain unmodified.

The "planes enabled for display bit mask" instruction specifies the planes that are to be enabled to index the color palette. All other planes give binary 0 values to the serializer and palette at all times.

Any non-existent bit planes specified in the masks are ignored.

The individual color masks are used when a RGB type palette is loaded, that is, a palette that allows separate bit fields within an eight-bit color index for each red, green and blue component. In this case the individual color masks allow the arithmetic mixes to operate on individual color components separately. So for example, it is possible to average two colors and get a meaningful result.

Usage Notes

- The HSBP order overrides all other controls. For example, if a current color of X'00000018' has been selected, but "planes enabled for update" has been set to X'00000010', then only plane 4 is altered by a "draw" instruction in the current color.

- The HSBP order is **not** valid between an HBAR and an HEAR order.
- To clear the individual color masks, they must all be set to an X'FFFFFFFF'.
- The ability to set color masks is an extension to the IBM Display Adapter 8514/A Base Function Set; it is supported by Extended Function Set (1) adapters only.

HQCOORD - Query Coordinate Types

Function: The HQCOORD order verifies support for a coordinate type.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (4)
2	BYTE	Format of each coordinate: Bits 7—4: Number of bytes Bits 3—0: Number of fractional bytes
3	BYTE	Format of each relative coordinate: Bits 7—4: Number of bytes Bits 3—0: Number of fractional bytes
4	BYTE	Number of dimensions (2)
5	BYTE	Return flags: Bit 7, Coordinate format: = 0: Supported = 1: Not supported Bit 6, Relative format: = 0: Supported = 1: Not supported Bit 5, Dimensions: = 0: Supported = 1: Not supported Bits 4—0: Reserved

Figure 3-38. HQCOORD: Query Coordinate Types

Description: The HQCOORD order examines the contents of the coordinate, relative coordinate, and dimension fields, and sets the appropriate flag, if the format is not supported by the adapter.

Usage Note: The HQCOORD order can be used to verify an HSCOORD order before the HSCOORD order is passed to the adapter.

HSCOORD - Set Coordinate Types

Usage Note: This order has no effect on the adapter.

HESC - Stop Processing (Escape)

Usage Note: This order has no effect on the adapter.

HSAFP - Set Area Fill Plane

Function: The HSAFP order specifies an address in personal computer storage that is to be used as an area fill plane.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (5)
2	DWORD	Address of area fill buffer
6	BYTE	Flags:
		Bits 7—0: Reserved (must be X'80')

Figure 3-39. HSAFP: Set Area Fill Plane

Description: The HSAFP order enables a software area fill plane for use. Typically, this order is needed when:

- An HSBMAP order has been issued and the application needs to use the area orders, markers, or graphics text.

See "HQMODE - Query Current Mode" on page 3-36 and "HQDPS - Query Drawing Process State Size" on page 3-55 for the size of area fill plane required with the current destination bitmap.

Usage Note: The area fill plane is also used as a graphics text cache, and marker cache. The address of the area fill buffer is a 32-bit offset from the start of display RAM and must be DWORD-aligned.

HQDPS - Query Drawing Process State Size

Function: The HQDPS order returns the size (in bytes) of these four elements of the drawing process:

- Task-dependent state area
- Driver stack usage
- Installed display direct access storage
- Area fill plane required.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (6 or 14)
2	WORD	Buffer size (bytes)
4	WORD	Stack usage (bytes)
6	WORD	Save palette buffer size (bytes)
8	DWORD	Size of installed display direct access storage (bytes)
12	DWORD	Size of area fill plane (bytes) required for current display mode

Figure 3-40. HQDPS: Query Drawing Process State Size

Size Fields: The data size fields contain the following information:

Field	Size requirement (bytes)
Buffer Size	Task-dependent state data length
Stack usage	Maximum adapter interface driver code stack
Save palette	Palette contents buffer

Usage Notes: Spare display direct access storage is calculated by the following process:

1. Use the HQDPS order to find out how much display storage is installed, and the size of the area fill plane required.
2. Use the HQBMAP order to calculate the storage needed by the current screen using the formula:

   Storage (bytes) = (width * height * bits/pel value) ÷ 8
3. Add the size of the area fill plane found at (1) to the number calculated at (2) to find the address (from the start of display RAM) of the available storage.

4. Subtract the address found at (3) from the total storage found at (1) to find the available display direct access storage.

When the HQDPS order is executed before an HOPEN is performed, all the fields returned are valid if the adapter interface code is already installed.

The size of installed display direct access storage and the size of area fill plane, is an extension to the IBM Display Adapter 8514/A Base Function Set; it is supported by Extended Function Set (1) adapters only.

See "State Data" on page 2-13 for a description of states.

Attribute Orders

HSMARK - Set Marker Shape

Function: The HSMARK order defines the shape of the current marker symbol.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥10)
2	BYTE	Cell width in pels (cx)
3	BYTE	Cell height in pels (cy)
4	BYTE	Flags: Bit 7: = 0: Monochrome = 1: Multicolor Bit 6: = 0: Byte/pel format = 1: Packed format Bits 5—0: Reserved (must be 0)
5	BYTE	Reserved (must be 0)
6	WORD	Length of image definition (bytes)
8	DWORD	Address of marker image definition
12	DWORD	Address of marker color definition

Figure 3-41. HSMARK: Set Marker Shape

Description: The HSMARK order sets the current marker symbol to a specified image.

The marker image is a rectangular bit pattern that starts at the top left corner with bit 7 of byte 0, and continues from left to right in descending rows. There are no padding bits between rows.

In byte format, marker color is defined by a list of one-byte color indices.

In packed format, marker color is defined by "packed" data. For example, at 2 bits/pel resolution, there are four color indices to a byte.

Packed format is an extension to the IBM Display Adapter 8514/A Base Function Set; it is supported by Extended Function Set (1) adapters only.

Note: A color index must always match the destination bitmap with which it is going to be used.

There is a color index for every pel of the image, including background (0) pels. Each color index is applied in turn to each pel in the image definition.

If the "monochrome" flag is selected, the marker is drawn in the current foreground color and mix on the current background color and mix.

If the "multicolor" flag is selected, the marker is drawn in the foreground colors extracted from the marker color definition, on the current background color and mix.

The address of the marker color definition must be supplied, and the order length must be fourteen bytes.

Usage Note: If packed format is specified for the color definition, the bits/pel values of the color data must match those of the destination bitmap, because color expansion is not supported.

HSPATT - Set Pattern Shape

Function: The HSPATT order defines the shape of the current area fill pattern symbol.

Byte	Content	Meaning
0	WORD	Length of following data (≥10)
2	BYTE	Cell width in pels (cx)
3	BYTE	Cell height in pels (cy)
4	BYTE	Flags: Bit 7: = 0: Monochrome = 1: Multicolor Bit 6: = 0: Byte/pel format = 1: Packed format Bits 5—0: Reserved (must be 0)
5	BYTE	Reserved (must be 0)
6	WORD	Length of image definition (bytes)
8	DWORD	Address of pattern image definition
12	DWORD	Address of pattern color definition

Figure 3-42. HSPATT: Set Pattern Shape

Description: The HSPATT order sets the current pattern symbol to a specified image.

The pattern image is a rectangular bit pattern that starts at the top left corner with bit 7 of byte 0, and continues from left to right in descending rows. There are no padding bits between rows.

In byte/pel format, pattern color is defined by a list of 1-byte color indices.

If pattern color is defined by "packed" data; for example, at 2 bits per pel, there are four color indices to a byte.

Packed format is an extension to the IBM Display Adapter 8514/A Base Function Set; it is supported by Extended Function Set (1) adapters only.

There is a color index for every pel of the pattern, including background pels. Each color index is applied in turn to each pel in the image definition.

If the "monochrome" flag is selected, the pattern is drawn in the foreground color and mix for ones, and on the background pattern and mix for zeros.

If the "multicolor" flag is selected, then when the pattern bit is a 1, the pattern is drawn in the corresponding color extracted from the pattern color definition, using the current foreground mix. When the pattern bit is 0, the pattern is drawn using the current background color and mix.

Usage Notes

- During a patterned area fill, the pattern position within the area is controlled by the HSPATTO order.
- To select solid fill, a 1 × 1 pattern cell size must be specified.
- If packed format for the color definition is specified, the bit/pel value of the color data must match the bit/pel value of the destination bitmap, because color expansion is not supported.

HSPATTO - Set Pattern Reference Point

Function: The HSPATTO order sets the reference point or origin for area fill pattern symbols.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (p)
2	P0	Pattern reference point

Figure 3-43. HSPATTO: Set Pattern Reference Point

Description: The pattern reference point is set to that specified in the order.

Usage Note: The pattern reference point specifies the top left corner location of an instance of the pattern cell.

HSLT - Set Line Type

Function: The HSLT order sets the current line type to the value specified.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥1)
2	BYTE	Line type value: X'00': Load user line type X'01': Dotted line X'02': Short dashed line X'03': Dash-dot line X'04': Double dotted line X'05': Long dashed line X'06': Dash-double-dot line X'07': Solid line X'08': Invisible line X'09': Dotted line 2
3	BYTE	Reserved
4	DWORD	Address of user line type definition

Figure 3-44. HSLT: Set Line Type

Description: Line type is set to the value specified in the order.

The user line type data is a word byte count followed by a list of byte pairs. The first byte of each pair gives the number of screen space pels "on" (foreground); the second byte gives the number of screen space pels "off" (background).

The maximum pattern run length allowed is 48 pels (in user line definition).

The pel "on—off" repetition sequences for the various line types are as follows:

Line Type	Pel Sequence
Dotted	1 on, 2 off
Short dashed	5 on, 3 off
Dash-dot	6 on, 4 off, 2 on, 4 off
Double dotted	2 on, 4 off, 2 on, 8 off
Long dashed	9 on, 3 off
Dash-double-dot	8 on, 4 off, 2 on, 4 off, 2 on, 4 off
Dotted (2)	1 on, 1 off

Usage Note: The HSCP, HCLINE, and HCRLINE orders inherit the line pattern count from previous orders. HSLT and all other line orders reset the line pattern count.

The Dotted (2) line type is an extension to the IBM Display Adapter 8514/A Base Function Set; it is supported by Extended Function Set (1) adapters only.

HSLW - Set Line Width

Function: The HSLW order sets the current line width value.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (1)
2	BYTE	Line width value

Figure 3-45. HSLW: Set Line Width

Description: The HSLW order sets line width to the value specified.

Usage Notes

- The adapter interface supports 1-pel or 3-pel line widths, according to the value set in the HSLW order, as follows:

Value	0	1	>1
Line width (pels)	1	1	3

- Lines 3 pels wide are thickened by drawing an extra line to the left and right (Y-major) or below and above (X-major).
- The line width value is an unsigned integer; that is, there are no negative line widths.

HSCOL - Set Color

Function: The HSCOL order sets the foreground color index to the value specified.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (4)
2	DWORD	Color index

Figure 3-46. HSCOL: Set Color

Description: The current foreground color index is set to the value specified in the order.

Usage Notes

- See "Bitplane and Bitmap Models" on page 2-3 for a description of the bit plane model.
- The reserved color X'FFFFFFFF' is used to enable multi-plane character sets. See "Programmable Character Set Specification" on page 3-70.
- Only the least significant 8 bits of the color index are used.

HSBCOL - Set Background Color

Function:

The HSBCOL order sets the specified background color index value.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (4)
2	DWORD	Color index

Figure 3-47. HSBCOL: Set Background Color

Description: The current background color index is set to the value specified in the order.

Usage Notes

- See "Bitplane and Bitmap Models" on page 2-3 for a description of the bit plane model.
- The reserved color X'FFFFFFFF' is used to enable multi-plane character sets. See "Programmable Character Set Specification" on page 3-70.
- Only the least significant 8 bits of the color index are used.

HSMX - Set Mix

Function: The HSMX order sets the specified value of current mix.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (2)
2	BYTE	Foreground mix value: X'00': Retain previous mix X'01': OR X'02': Overpaint X'03': Reserved X'04': Exclusive OR X'05': Leave alone X'06': Maximum X'07': Minimum X'08': Add (screen + new) X'09': Subtract (screen – new) X'0A': Subtract (new – screen) X'0B': Average (screen + new) ÷ 2 X'0C': Reserved : X'0F': Reserved X'10': zero X'11': screen and new X'12': ¬screen and new X'13': new X'14': screen and new X'15': screen X'16': screen xor new X'17': screen or new X'18': ¬screen and new X'19': ¬screen xor new X'1A': ¬screen X'1B': ¬screen or new X'1C': ¬new X'1D': screen or ¬new X'1E': ¬screen or ¬new X'1F': one X'20': Reserved : X'FF': Reserved
3	BYTE	Background mix value (same as foreground mix)

Figure 3-48. HSMX: Set Mix

Description: The foreground and background mixes are set to the values specified in the order.

Usage Notes

- The arithmetic mix functions (maximum, minimum, add, subtract, average) operate on a contiguous set of graphics planes. Planes that are disabled for update take no part in the mix. If non-contiguous planes are enabled, the results are undefined.
- The results yielded by each mix function, expressed as the binary value of the plane color index (A) and a new color (B), are as follows:

Mix	Yield
Maximum	Maximum binary value of A and B
Minimum	Minimum value of A and B
Add	Sum of A + B: an "overflow" result is clipped to the maximum binary value
Subtract	Difference between A and B: a result less than zero is clipped to zero
Average	Half the binary sum of A + B

- The arithmetic mix functions are subject to the current values of red, blue, and green masks, as set by the HSBP order. This means that the individual red, green, and blue fields within pels can operate independently in arithmetic mix operations; it is, for example, possible to "average" their values.

HSCMP - Set Color Comparison Register

Function: The HSCMP order sets the value of the color comparison register.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (5)
2	DWORD	Comparison color index
6	BYTE	Logic function:
		X'00': True
		X'01': Plane data > comparison
		X'02': Plane data = comparison
		X'03': Plane data < comparison
		X'04': False
		X'05': Plane data ≥ comparison
		X'06': Plane data ¬ = comparison
		X'07': Plane data ≤ comparison

Figure 3-49. HSCMP: Set Color Comparison Register

Description: The HSCMP sets the color comparison register to the required color and logic values.

The plane data is compared with the color register. If the comparison result is "true," the existing bit plane data is left unmodified; if the result is "false," the current foreground or background mix is used.

Only those planes enabled for update take part in the comparison.

Usage Notes

- The color comparison register can be used to implement "underpaint" if it is set to the color of the erased screen.
- The color comparison register has uses in image processing.
- The color comparison function does not apply to alphanumeric orders.
- The HSCMP order is **not** valid between an HBAR and an HEAR order.

Programmable Character Set Specification

Character definitions are either rectangular, single- or tri-plane images, or short stroke vectors.

Each character set has a "definition block" in the following format:

Character Set Definition Block

Byte	Content	Meaning
0	BYTE	Reserved
1	BYTE	Type of character set: 0: Image or multiplane image 1: Reserved 2: Reserved 3: Short stroke vector
2	BYTE	Reserved
3	DWORD	Reserved
7	BYTE	Cell width in pels (cx)
8	BYTE	Cell height in pels (cy)
9	BYTE	Reserved
10	WORD	Character definition cell size, bytes
12	WORD	Flags: Bit 15: Reserved (must be 0) Bit 14: = 0: Single plane = 1: Multiplane Bit 13: = 0: Not proportionally spaced = 1: Proportionally spaced Bits 12—0: Reserved (must be 0)
14	DWORD	Address of index table
18	DWORD	Address of character envelope table
22	BYTE	Initial code point
23	BYTE	Final code point
24	DWORD	Address of character definition table
28	WORD	Reserved
30	DWORD	Address of second character definition table
34	WORD	Reserved
36	DWORD	Address of third character definition table

Figure 3-50. The Character Set Definition Block

Image Characters: Each character code point is used to index the image index table, which starts at the initial code point and ends at the final code point. If an attempt is made to draw a character not defined in the index table, the character referred to by the initial code point is drawn instead.

Image Index Table: The index table entry defines the location as follows:

Byte	
0—1	Offset of character definition
2	

Figure 3-51. The Image Index Table Entry

The address of the character definition is calculated by adding the unsigned offset field to the address of the character definition table.

Character Definition Cell Size: Each character definition starts on a byte boundary and occupies a whole number of bytes (m), given by:

$$m = [(cx \star cy) + 7] \div 8$$

when:

cx = cell width, pels
cy = cell height, pels

Each image character definition is in the form of an array of bits. Each bit corresponds to one pel. A pel takes the current foreground color, if its corresponding bit is a binary 1, or the background color, if its corresponding bit is a binary 0.

Image character definition in a character cell starts at the top left pel and finishes at the bottom right pel.

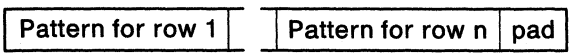

Figure 3-52. Image Character Definition

Multiplane Image Characters: Multiplane image characters consist of three monochrome images, drawn as follows:

- If the current character foreground color is not X'FFFFFFFF':

  The three images are mixed with the OR logic function to create a single monochrome image. The image is drawn with the current foreground color and mix on the current background color and mix.

- If the current character foreground color is the reserved value X'FFFFFFFF':

  For each pel in the image character, a color index is obtained from the multiplane text color index table. See "HXLATE - Assign Multiplane Text Color Index Table" on page 3-78.

Short Stroke Vector Characters: The index table is used to locate each character definition within the character definition block. Each character code point is used to index the table. The index table starts at the initial code point and ends at the final code point. If an attempt is made to draw a character not defined in the index table, the character referred to by the initial code point is drawn instead.

Short stroke vector definition in a character cell must start at the bottom left pel, and end one pel to the right of the bottom right pel.

The index table entry defines the location as follows:

Byte	Short stroke vectors
0—1	Offset of character definition
2	

Figure 3-53. The Short Stroke Vector Index Table Entry

The address of the character definition is calculated by adding the unsigned offset field to the address of the character definition table.

Short stroke vector character definitions are in the form of an array of short drawing orders. Each order is 1 byte long.

The end of a short stroke definition is defined by a X'00'.

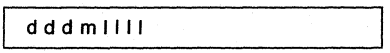

Figure 3-54. Short Stroke Vector Definition

The bit setting "d d d" gives the vector direction in degrees counter-clockwise from the horizontal:

d d d	Direction
0 0 0	0°
0 0 1	45°
0 1 0	90°
0 1 1	135°
1 0 0	180°
1 0 1	225°
1 1 0	270°
1 1 1	315°

The "m" bit setting is the "draw/move" flag:

m = 0 = move
m = 1 = draw

The "l l l l" bit setting gives the projection of line length on the horizontal or vertical axis (depending on the direction set by "d d d") in screen space pels. The last pel is always null. The binary value 0 0 0 0 performs no operation. In "draw" mode, a binary value of n writes n pels, and moves the drawing position one pel further.

Character Envelope Table: The character envelope table is used to provide proportional spacing, and only needs to be supplied if proportional spacing has been selected.

Each character code point is used to index the character envelope table. The character envelope table starts at the initial code point and ends at the final code point.

Each entry in the character envelope table defines a left and a right margin (in screen space pels).

See "Short Stroke Vector Characters" on page 3-72.

Byte	
0	Left margin
1	Right margin
2	

Figure 3-55. Character Envelope Table Entry

The left margin is subtracted from the "horizontal character start" position before the character is drawn; the right margin is subtracted from the "horizontal character start" position after the character is drawn.

For image fonts, the left and right margins represent areas of the character definition that are not drawn on the screen.

Character Cell Size and Spacing: The character cell size is taken from the character set definition block. For proportionally-spaced characters, the width value "cx" in the definition block is decremented by a value given by the character envelope table.

In a string, the second and subsequent characters are drawn immediately to the right of the preceding character(s).

Character Orders

HSCS - Set Character Set

Function: The HSCS order sets the specified current character set.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (4)
2	DWORD	Address of the character set definition block

Figure 3-56. HSCS: Set Character Set

Description: The HSCS order changes the current character set.

The address of the character set definition is an address in the controlling environment. The controlling system must preserve the contents of the character set buffer until a subsequent HSCS order has been processed by the adapter.

The character set definition format is given in "Programmable Character Set Specification" on page 3-70.

If the character set definition is changed in any way, the HSCS order must be reissued.

HCHST - Character String at Given Position

Function: The HCHST order draws a character string at a given position.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥p+s)
2	P0	Coordinate data of point at which the "bottom left corner" of the character string is placed
2+p	STRING	List of code points in the string

Figure 3-57. HCHST: Character String at Given Position

Description: The HCHST order draws a character string at point P0. Each code point is one byte long.

Attributes

- Foreground and background color
- Color mix
- Color comparison.

Usage Notes

- The HCHST order is not valid between an HBAR and an HEAR order when it is used for multiplane fonts.
- For multiplane text, when the foreground color is set to X'FFFFFFFF', the background color and color mix attributes of the HCHST order are not applicable.

Graphics text is subject to control by the Scissor function.

The current position is set to P0.

HCCHST - Character String at Current Position

Function: The HCCHST order draws a character string at the current position.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥0)
2	STRING	List of code points in the string

Figure 3-58. HCCHST: Character String at Current Position

Description: The HCCHST order draws a character string at the current position. Each code point is one byte long.

Attributes

- Foreground and background color
- Color mix
- Color Comparison.

Usage Notes

- The HCCHST order is not valid between an HBAR and an HEAR order when it is used for multiplane fonts.
- For multiplane text, when the foreground color is set to X'FFFFFFFF', the background color and color mix attributes of the HCHST order are not applicable.

Graphics text is subject to control by the Scissor function.

The current position is unchanged.

HXLATE - Assign Multiplane Text Color Index Table

Function: The HXLATE order provides a color index translate table for use with multiplane text orders.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (32)
2	DWORD	Translate table, entry 1
.		
.		
.		
30		Translate table, entry 8

Figure 3-59. HXLATE - Assign Multiplane Color Index Table

Description: The HXLATE order is used when processing the orders that define a multiplane image set (HCHST and HCCHST). The HXLATE order provides a translation table from which the color index, used for each pel of the character, can be obtained. The combination of the pel values in each of the three plane definitions is used to address the table.

The table address is given by:

[(pd1) 2 * * 0] + [(pd2) 2 * * 1] + [(pd3) 2 * * 2]

where:

pd1 = plane 1 pel value (0 or 1)

pd2 = plane 2 pel value (0 or 1)

pd3 = plane 3 pel value (0 or 1).

Alphanumeric Orders

These orders allow the user to display alphanumerics that are constrained by character cells. Character format is more rigid than that of graphics text; characters are arranged in rows and columns, and have attributes associated with them.

Each character is defined by a pattern of foreground and background bits and a set of character attributes. A single-byte code point is used to select the character shape from one of four symbol sets. The character attributes are:

- Foreground color (1 of 16)
- Background color (1 of 16)
- Reverse video
- Underscore
- Overstrike
- Background (transparent or opaque)
- Font (one of four).

There is an alphanumeric cursor that marks one character cell. The following cursor options are supported:

- Hidden
- Normal (variable size)
- Left arrow
- Right arrow.

Note: Alphanumeric characters are written into the same bit planes as the graphics, and it is the responsibility of the controlling environment to:

- Manage the alpha-graphic interaction.
- Save background information to enable transparency to work.

ABLOCKMFI - Write Character Block (MFI)

Function: The ABLOCKMFI order writes a block of characters to the bit planes in Mainframe Interactive (MFI) mode.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (10)
2	BYTE	Start column (0 to n)
3	BYTE	Start row (0 to n)
4	BYTE	Number of character cells "across"
5	BYTE	Number of character cells "down"
6	DWORD	Start address of character block
10	BYTE	Width of character buffer
11	BYTE	Reserved (must be 0)

Figure 3-60. ABLOCKMFI: Alphanumeric Character Block (MFI)

Description: The character string located by the "start address of character block" field is drawn in the destination bitmap, beginning with the character cell at the "start" column and row.

Subsequent characters are placed, in order of increasing column addresses, until the specified number of character cells "across" (that is, in the horizontal axis) has been written. The row count then increments to the next row and the start column is reset to specified start column. The next character cell address in the linear array is calculated by adding the width of the character buffer (multiplied by four) to the buffer address of the current line.

Character cells are addressed from the top left (0,0). Column addresses increment to the right, and row addresses increment downwards.

The character block is a linear array of character definitions. Each character is represented by a 4-byte field as shown in the following figure.

Byte	Content	Meaning
0	BYTE	Character code
1	BYTE	Color attribute: Bits 7—4: Background color Bits 3—0: Foreground color
2	BYTE	Highlight attribute: Bit 7: Underscore Bit 6: Reverse video Bit 5: Overstrike Bit 4: Background: = 0: Opaque = 1: Transparent Bit 3: Reserved (must be 0) Bit 2: Spare Bits 1—0: Font number
3	BYTE	Reserved

Figure 3-61. The Adapter Character Representation (MFI)

Optimization: The AERASE and ASCROLL orders must be used for erasing, scrolling, and inserting data on the screen to obtain the best performance.

Usage Note: The Scissor function is applied to alphanumeric as well as graphic updates.

ABLOCKCGA - Write Character Block (CGA)

Function: The ABLOCKCGA order is similar to that for ABLOCKMFI, except that it supports a 2-byte character attribute sequence for Color Graphics Adapter (CGA) operation.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (10)
2	BYTE	Start column (0 to n)
3	BYTE	Start row (0 to m)
4	BYTE	Number of character cells "across"
5	BYTE	Number of character cells "down"
6	DWORD	Start address of character block
10	BYTE	Width of character buffer
11	BYTE	Highlight attribute for block

Figure 3-62. ABLOCKCGA: Alphanumeric Character Block (CGA)

Description: The character string located by the "address of character block" field is written in the destination bitmap, beginning with the character cell at the "start" column and row. Subsequent characters are placed, in order of increasing column addresses, until the specified number of character cells "across" (that is, in the horizontal axis) has been written. The row count then increments to the next row and the start column is reset. The next character cell address in the linear array is calculated by adding the width of the character buffer (multiplied by two) to the buffer address of the current line.

Character cells are addressed from the top left (0,0). Column addresses increment to the right, and row addresses increment downwards.

The character block is a linear array of character definitions. Each character is represented by a 2-byte field as shown in Figure 3-63 on page 3-83.

The highlight attribute byte in the parameter block is of the same format as the highlight attributes in the ABLOCKMFI order.

Byte	Content	Meaning
0	BYTE	Character code
1	BYTE	Color attribute: Bits 7—4: Background color Bits 3—0: Foreground color

Figure 3-63. The Adapter Character Representation (CGA)

Optimization: The AERASE and ASCROLL orders must be used for erasing, scrolling, and inserting data on the screen to obtain the best performance.

Usage Note: The Scissor function is applied to alphanumeric, as well as graphics, updates.

ASCELL - Set Alpha Cell Size

Function: The ASCELL order sets the cell size for alphanumeric operations.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (2)
2	BYTE	Cell width, pels
3	BYTE	Cell height, pels

Figure 3-64. ASCELL: Set Alpha Cell Size

Description: The ASCELL order sets the alpha cell size in the task-dependent state. This size setting overrides the default cell size associated with the HOPEN or HSMODE orders, that store the cell size in the task-independent state.

Usage Notes

- If either width or height is zero, the defaults in the task-independent state will be used.
- Alphanumeric overstrike is not supported if ASCELL is active.

AERASE - Erase Rectangle

Function: The AERASE order sets a rectangle of character cells to a specified background color.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (5)
2	BYTE	Starting column (0 to n)
3	BYTE	Starting row (0 to m)
4	BYTE	Number of character cells in horizontal axis "across"
5	BYTE	Number of character cells in vertical axis "down"
6	BYTE	Color: Bits 7—4: Background color Bits 3—0: Reserved = B'0000'

Figure 3-65. AERASE: Erase Rectangle

Description: The AERASE order sets a defined rectangle of character cells to the background color specified in the color field. The rectangle's position and size are defined as follows:

- The top left corner is given by the starting row and starting column fields.
- The size (in character cells) is given by the numbers of cells in the "across" and "down" fields.

ASCROLL - Scroll Rectangle

Function: The ASCROLL order copies a rectangle of character cells on the screen.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (6)
2	BYTE	Starting column (0 to n) of source
3	BYTE	Starting row (0 to m) of source
4	BYTE	Number of character cells in horizontal axis "across"
5	BYTE	Number of character cells in vertical axis "down"
6	BYTE	Starting column of destination
7	BYTE	Starting row of destination

Figure 3-66. ASCROLL: Scroll Rectangle

Description: The ASCROLL order copies a "source" rectangle to a "destination" rectangle. Both rectangles are defined by a top left corner position given in the source and destination starting column and starting row fields. The size of the source rectangle is given by the numbers of character cells "across" and "down."

Both source and destination rectangles must be entirely within the bitmap. The effect of copying a rectangle to an overlapping rectangle is as if the source data is stored in a separate buffer.

ACURSOR - Set Cursor Position

Function: The ACURSOR order sets the alphanumeric cursor position.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (2)
2	BYTE	Cursor position, column (0 to n)
3	BYTE	Cursor position, row (0 to m)

Figure 3-67. ACURSOR: Set Cursor Position

Description: The ACURSOR order removes the cursor from its previous character cell location, and draws it in the character cell defined by the cursor position "row" and "column" fields.

Usage Note: The alpha cursor is a display-only function which has no meaning in a non-screen bitmap.

ASCUR - Set Cursor Shape

Function: The ASCUR order sets the alphanumeric cursor shape.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (3)
2	BYTE	Cursor start line (0 to n)
3	BYTE	Cursor stop line (0 to n)
4	BYTE	Attribute: X'00': Normal X'01': Hidden X'02': Left arrow X'03': Right arrow

Figure 3-68. ASCUR: Set Cursor

Description: The cursor shape and attributes are set for subsequent cursor operations. Zero in the cursor start and stop lines (bytes 2 and 3) defines the top of the character cell.

Usage Notes

- After a mode change, the cursor is hidden.
- The cursor is drawn with an XOR mix, and all interaction between alpha and the alpha cursor is handled by the interface. However, any graphics corruption of the characters, the cursor, or both must be dealt with by the controlling environment.
- If the start line is greater than the stop line, no cursor is drawn.
- The start and stop line must be within the bounds of the character cell.
- If the start line is X'FF', the current start-stop line definition is used; that is, the current size is inherited.

ASFONT - Select Character Set

Function: The ASFONT order selects one of the four alphanumeric character sets available with the interface.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (6)
2	BYTE	Font number (0—3)
3	BYTE	Reserved
4	DWORD	Address of the character set definition block

Figure 3-69. ASFONT: Select Character Set

Description: The character set located at the character set definition address is used to write all subsequent alphanumeric characters with the specified font.

The character set definition format is given in "Programmable Character Set Specification" on page 3-70. The character set must be a single-plane image character set or a short stroke vector set.

Usage Note: The symbol set and cell size selected must match. Incorrectly matched symbol sets produce the following results:

- Short stroke vectors are drawn in the bottom left corner of the current character cell and can overlap if they are larger than the cell.
- If an incorrectly matched image character is drawn, the current cell is filled but its content is not predictable.

AXLATE - Assign Alpha Attribute Color Index Table

Function: The AXLATE order provides an attribute to the color index translate table.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (128)
2	DWORD	Foreground translate table entry 1
.		
.		
62	DWORD	Foreground translate table entry 16
66	DWORD	Background translate table entry 1
.		
.		
126	DWORD	Background translate table entry 16

Figure 3-70. AXLATE - Assign Alpha Color Index Table

Description: When processing the alphanumeric orders, the AXLATE order uses the character attributes to address the two color index tables and obtain values for use as follows:

Character attribute bits:	3—0	7—4
Color index table addressed:	Background	Foreground
Value obtained used for:	Character background	Character

Chapter 4. The Extended Function Sets

As described in Chapter 2, order definitions use the following symbols:

- **c** denotes the length (in bytes) of one absolute coordinate.
- **r** denotes the length (in bytes) of a pair of relative coordinates.
- **p** denotes the length (in bytes) of a pair of absolute coordinates.
- **s** denotes the length (in bytes) of a variable length string.

For the implementation described in this book, c=2, r=2, p=4, and s is variable.

Extended Function Set (1)

HDLINE - Disjoint Line

Function: The HDLINE order defines zero or more disconnected straight lines.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥p)
2	COORD	Coordinate data of first line start (P0)
2+p	COORD	Coordinate data of first line end (P1)
.		
.		
.		
2+np	COORD	Coordinate data of last line end (Pn)

Figure 4-1. HDLINE: Line from Given Position to Given Position

Description: A line is drawn from a point P0 to point P1, from P2 to P3, and from Pn-1 to Pn. Any number of pairs of start and end points can be present, bounded only by the maximum length specified by the order. Consecutive pairs of points in the order are joined by straight lines.

Usage Note: Current position is set to the last point specified.

Attributes

- Line type
- Line width
- Foreground and background color
- Color mix
- Color comparison.

Control

- "Last pel" processing affects the way in which lines are drawn, as described at "HSGQ - Set Graphics Quality" on page 3-40.
- Lines are subject to control by the Set Scissor function.
- HDLINE resets the line pattern count to zero for each line, so that the next line starts at the beginning of a line pattern.

HQDEVICE - Query Device Specific

Usage Note: This order has no effect on the adapter.

ASGO - Set Alpha Grid Origin

Function: The ASGO order changes the cell grid origin for alphanumeric operations.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (4)
2	WORD	Horizontal cell offset, pels
4	WORD	Vertical cell offset, pels

Figure 4-2. ASGO: Set Alpha Grid Origin

Description: This order sets the cell grid origin in the task-dependent state. This setting overrides the default origin associated with the HOPEN or HSMODE orders, that store the origin in the task-independent state.

Both the horizontal and vertical offsets are specified as numbers of display area pels before the start of the first character cell:

- Horizontal: from the left of the display area
- Vertical: from the top of the display area.

Both offsets are signed as follows:

- Positive: right or down
- Negative: left or up.

Usage Notes

- The grid origin is reset to 0,0 by any subsequent HSMODE orders.
- Offset distances are specified to the top left corner of the first cell.
- The ASGO order allows the alphanumeric orders ABLOCKMFI, ABLOCKCGA, and AERASE to write or erase character strings at any pel position.
- To disable a previously set alpha grid origin, this order must be called with horizontal and vertical cell offsets both 0.

HPEL - Write Pel String

Function: The HPEL order writes a string (zero, one, or more) of pels from left to right horizontally.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (≥2+p)
2	P0	Coordinate data of first pel
2+p	WORD	Pel count
2+n(p+2)	Pn	Coordinate data of first pel of pel run n
(n+1)(p+2)	WORD	Pel count of pel run n

Figure 4-3. HPEL: Write Pel String

Description: The HPEL order writes a number of pels, given by pel count, at the given start position, from left to right across the screen. Several pel runs can be written by the same order, by including several sets of start positions and pel counts.

Usage Notes

- The HPEL order writes single pels or horizontal lines, either singly or in multiples, and it can therefore be used to draw pre-scanned areas as in "flood fill."
- HPEL is **not** valid in an area definition.
- HPEL does not reset, nor is it affected by, the line pattern.
- HPEL is not affected by the line width.
- HPEL is a faster way of drawing solid horizontal lines than the line orders.
- HPEL can be used to write individual pels.

Attributes

- Foreground and background color
- Color mix
- Color comparison.

Control

- The HPEL order is subject to control by the Set Scissor function.
- "Last pel null" is **not** valid for HPEL.
- HPEL does not affect the current position.

HRPEL - Read Pel String

Function: The HRPEL order reads a string (zero, one, or more) of pels from left to right horizontally, starting at a given position.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (6+p)
2	DWORD	Address of buffer for data read
6	P0	Coordinate data of first pel
6+p	WORD	Pel count

Figure 4-4. HRPEL: Read Pel String at Given Position

Description: The HRPEL order reads a number of pels, given by pel count, from the given start position, from left to right across the screen. The color indexes read for each pel are placed in the buffer supplied, in a "packed" format.

The order of pels within a byte is from high to low: for example, reading one pel from a four-plane destination bitmap puts the pel value in the high portion of the byte.

The HRPEL read operation is not affected by attributes or interface control functions. The HRPEL order does not affect the current position.

Usage Note: The HRPEL order can be used to read a single pel or a string of pels; the latter function is useful for scanning prior to "flood fill."

The maximum number of bytes that can be read is 16KB (KB equals 1024 bytes).

HPSTEP - Plot and Step

Function: The HPSTEP order defines a series (zero, one, or more) of adjacent pel runs, starting from at the given position.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (4+p or 8+p)
2	P0	Coordinate data of first pel
2+p	DWORD	Address of plot and step definition buffer
6+p	DWORD	Address of source data buffer

Figure 4-5. HPSTEP: Plot and Step

Description: Each pel run is drawn, starting from a given position, according to the data held in the plot and step definition buffer. The plot and step definition buffer contains bytes coded as follows:

Bits 7—5: Direction of the step (d d d)

Bit 4: Plot and step flag (f)

Bits 3—0: Pel run length (r r r r).

Coding d d d gives the step direction in degrees counter-clockwise from the horizontal:

d d d	Direction
0 0 0	= 0°
0 0 1	= 45°
0 1 0	= 90°
0 1 1	= 135°
1 0 0	= 180°
1 0 1	= 225°
1 1 0	= 270°
1 1 1	= 315°

Code f sets the "plot and step" flag:

f = 1 = plot pel at current position and step to new position
f = 0 = step to next position without plotting

Coding r r r r sets the pel run length: $0 \leq r\ r\ r\ r < 16$.

The data in the plot and step definition buffer is terminated by a byte of zeros.

Variable data can also be used to supply color indexes for each pel. This data is held in the source data buffer. The bit/pel values of this variable data must match those of the current bitmap.

When this variable data is supplied, a color index is written to the current bitmap (under the current mix) for every pel in the pel data. The variable data held in the buffer is "packed," that is, not padded to bytes.

The number of color indexes in the variable data buffer must match the number of pels in the pel data. If no variable data is supplied (length field = 4+p), each pel is written using the current foreground color and mix.

Usage Notes

- The HPSTEP order can be used for area boundary definition, but:
  - HPSTEP causes an autoclose to any previously defined boundary if the autoclose function is enabled, and
  - source data is ignored.
- HPSTEP is affected by current line type but **not** by current line width.
- HPSTEP is always "last pel null."
- When variable data is supplied, and a step pel run is executed, the source data pointer is incremented by the number of pels stepped. Therefore, source data corresponding to a sequence of "plot and step" pel runs must be padded out with zero values when steps occur.

  The maximum size of the source data buffer is 4095 bytes.

Attributes

- Line pattern (HPSTEP resets the line pattern count)
- Foreground and background color
- Color mix
- Color comparison.

The order is subject to control by the Scissor function.

The order changes current position to the last point specified.

HCPSTEP - Plot and Step at Current Position

Function: The HCPSTEP order defines a series (zero, one, or more) of adjacent pel runs, starting at the current position.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (4 or 8)
2	DWORD	Address of plot and step definition buffer
6	DWORD	Address of source data buffer

Figure 4-6. HCPSTEP: Plot and Step at Current Position

Description: Each pel run is drawn, starting from a given position, according to the data held in the plot and step definition buffer. The plot and step definition buffer contains bytes coded as follows:

Bits 7—5: Direction of the step (d d d)

Bit 4: Plot and step flag (p)

Bits 3—0: Pel run length (r r r r).

Coding d d d gives the step direction in degrees counter-clockwise from the horizontal:

d d d	Direction
0 0 0	= 0°
0 0 1	= 45°
0 1 0	= 90°
0 1 1	= 135°
1 0 0	= 180°
1 0 1	= 225°
1 1 0	= 270°
1 1 1	= 315°

Code p sets the "plot and step" flag:

p = 1 = plot pel at current position and step to new position
p = 0 = step to next position without plotting

Coding r r r r sets the pel run length: $0 \leq r\ r\ r\ r < 16$.

The data in the plot and step definition buffer is terminated by a byte of zeros.

Variable data can also be used to supply color indexes for each pel. This data is held in the source data buffer. The bit/pel values of this variable data must match those of the current bitmap.

When this variable data is supplied, a color index is written to the current bitmap (under the current mix) for every pel in the pel data. The variable data held in the buffer is "packed," that is, not padded to bytes.

The number of color indexes in the variable data buffer must match the number of pels in the pel data. If no variable data is supplied (length field = 4+p), each pel is written using the current foreground color and mix.

Usage Notes

- The HCPSTEP order can be used for area boundary definition, but source data is ignored.
- HCPSTEP is affected by current line type but **not** by current line width.
- HCPSTEP is always "last pel null."
- When variable data is supplied, and a step pel run is executed, the source data pointer is incremented by the number of pels stepped. Therefore, source data corresponding to a sequence of "plot and step" pel runs must be padded out with zero values when steps occur.

  The maximum size of the source data buffer is 4095 bytes.

Attributes

- Line pattern (HCPSTEP does not reset the line pattern count)
- Foreground and background color
- Color mix
- Color comparison.

The order is subject to control by the Set Scissor function.

The order changes current position to the last point plotted.

HRSTEP - Read and Step

Function: The HRSTEP order reads a series (zero, one, or more) of adjacent pel runs.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (8+p)
2	P0	Coordinate data of first pel
2+p	DWORD	Address of read and step definition buffer
6+p	DWORD	Address of target data buffer

Figure 4-7. HRSTEP: Read and Step

Description: Each pel run is read, starting from a given position, according to the data held in the read and step definition buffer. The read and step definition buffer contains bytes coded as follows:

Bits 7—5: Direction of the step (d d d)

Bit 4: Read and step flag (f)

Bits 3—0: Pel run length (r r r r).

Coding d d d gives the step direction in degrees counter-clockwise from the horizontal:

d d d	Direction
0 0 0	= 0°
0 0 1	= 45°
0 1 0	= 90°
0 1 1	= 135°
1 0 0	= 180°
1 0 1	= 225°
1 1 0	= 270°
1 1 1	= 315°

Code f sets the "read and step" flag:

f = 1 = read pel at current position and step to new position
f = 0 = step to next position without reading

Coding r r r r sets the pel run length: $0 \leq r\ r\ r\ r < 16$.

The data in the read and step definition buffer is terminated by a byte of zeros.

The color indices read are held in the target data buffer in "packed" format. The bit/pel values of the color index data must match those of the current bitmap.

Attributes do not affect the HRSTEP read operation.

The HRSTEP order does not affect the current position.

Usage Note: When a step pel run is made, the pointer to the target definition buffer is moved by the number of pels stepped. The target buffer values moved over are left unchanged.

The maximum size of the target data buffer is 4095 bytes.

HRWVEC - Read or Write Vector

Function: The HRWVEC order supports a read or write vector drawing function with color data.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data [6+(np)]
2	BYTE	Flags: Bit 7: = 0: Read vector = 1: Write vector Bits 6—0: Reserved (must be 0)
3	BYTE	Reserved (must be 0)
4	DWORD	Address of data buffer
8	P0	Coordinate line start
8+p	P1	Coordinate of first line end
.		
.		
.		
8+np	Pn	Coordinate of n^{th} line end

Figure 4-8. HRWVEC: Read or Write Vector

Description: The HRWVEC order allows data to be read from, or written to, the destination bitmap under the control of a line algorithm.

Reading (Flag Bit 7 = 0): Data is read, starting at the line start coordinate P0 and continuing through the polyline coordinate list (P1—Pn) using a line drawing algorithm. The line data read is held, in "packed" format, in the user-specified source data buffer.

Writing (Flag Bit 7 = 1): Data from the source data buffer is written into the destination bitmap, starting at P0 and continuing through the polyline coordinate list (P1—Pn). The data is written under the control of a line algorithm, and mixed with the destination data according to its source value. Source zeros denote background mix; all other source values use foreground mix.

Usage Notes

- The maximum buffer length for line data in HRWVEC is 4095 bytes.
- The HRWVEC order is always "last pell null."

Attributes

- Color mix
- Color comparison.

Control: The HRWVEC order is subject to control by the Set Scissor function during write operations.

The HRWVEC order sets the current position to the last point specified for a "write" operation. Current position is unaffected by a "read" operation.

HSFPAL - Save Full Palette

Function: The HSFPAL order saves the total contents of the color palette and the display mask.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (8)
2	WORD	Format (=8)
4	DATA	Buffer

Figure 4-9. HSFPAL: Save Full Palette

Description: This order saves the contents of the palette and the display mask that can later be restored using the HRFPAL order. The length should be set to the buffer size returned by the HQDPS order + 2.

Usage Notes

- The HSFPAL order, together with the HRFPAL order, can be used to preserve palette integrity during operations that may corrupt it.
- These are adapter-specific orders and must be used in preference to HSPAL/HRPAL that save and restore only the high-order 6 bits of the palette data (resolution).

HRFPAL - Restore Full Palette

Function: The HRFPAL order restores the total contents of the color palette and the display mask.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data
2	WORD	Format (=8)
4	DATA	Buffer

Figure 4-10. HRFPAL: Restore Full Palette

Description: This order restores the contents of the palette and the display mask, as saved previously by the HSFPAL order. The length should be set to the buffer size returned by the HQDPS order + 2.

Usage Notes

- The HRFPAL order, together with the HSFPAL order, can be used to preserve palette integrity during operations that may corrupt it.
- These are adapter-specific orders and must be used in preference to HSPAL/HRPAL that save and restore only the high-order 6 bits of the palette data (resolution).

HSBMAP - Set Bitmap Attributes

Function: The HSBMAP order sets the current bitmap, that then becomes the destination for all subsequent drawing primitives.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (10)
2	BYTE	Flags: Bit 7: = 1: Reserved (must be 1) Bit 6: = 0: Not screen bitmap = 1: Screen bitmap Bits 5—0: Reserved (must be 0)
3	BYTE	Format, bits per pel: X'01': 1 X'02': 2 X'04': 4 X'08': 8
4	DWORD	Address of bitmaps
8	WORD	Width (in pels) of the bitmaps
10	WORD	Height (in pels) of the bitmaps

Figure 4-11. HSBMAP: Set Bitmap Attributes

Description: The HSBMAP order sets the destination bitmap for drawing operations.

Usage Notes

- The current position is set to 0,0.
- The HSBMAP order invalidates the area boundary definition and the default area fill plane allocation.
- The address is a 32-bit offset from the start of display RAM.
- A Screen bitmap must start on a QWORD boundary, and the width of the bitmap must be an integer multiple of QWORD's wide. That is, the result of the division, map width/pels per byte, must be an integer multiple of 4.
- A Not screen bitmap must start on a DWORD boundary, and the width of the bitmap must be an integer multiple of DWORD's wide. That is, the result of the division, map width/pels per byte, must be an integer multiple of 2.

- The Scissor is reset to the size of the bitmap.
- The order sets the update masks (see HSBP) to all bits (planes) enabled for update and display.
- The HSBMAP order does not clear the bitmap; the controlling environment should clear the bitmap, if necessary, using the HEGS order.
- The maximum size of a bitmap is limited by the size of available display RAM. The maximum width and height is 2048 pels.

HQBMAP - Query Bitmap Attributes

Function: The HQBMAP order returns the attributes of the current bitmap in the parameter block provided.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (10 or 14+p)
2	BYTE	Flags: Bit 7: Reserved (must be 1) = 1: Display direct access storage bitmap Bit 6: = 0: Not screen bitmap = 1: Screen bitmap Bits 5—0: Reserved (must be 0)
3	BYTE	Format, bits per pel: X'01': 1 X'02': 2 X'04': 4 X'08': 8
4	DWORD	Address of bitmap
8	WORD	Width (in pels) of the bitmap
10	WORD	Height (in pels) of the bitmap
12	P0	Coordinate of display window origin (top left corner)
12+p	WORD	Display window width
14+p	WORD	Display window height

Figure 4-12. HQBMAP: Query Bitmap Attributes

Description: This order returns the attribute information of the current bitmaps. If the current bitmap is the display RAM screen bitmap and the length of the parameter block has been set to (14+p), the window data is also returned.

HBMC - Bitmap Copy

Function: The HBMC order copies a block within the current bitmap, or from bitmap to bitmap.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (=36+3p)
2	WORD	Flags: Bits 15—14, Destination bitmap: = 00: Reserved = 01: Reserved = 10: VRAM = 11: Reserved Bits 13—12, Source bitmap: = 00: Not present = 01: Reserved = 10: VRAM = 11: Reserved Bits 11—10, Pattern bitmap: = 00: Not present = 01: Reserved = 10: VRAM = 11: Reserved Bit 9 = 1: Invert bitmap Bits 8—0, Reserved (must be 0)
4	WORD	Width in pels of block
6	WORD	Height in pels of block
8	BYTE	Format of destination bitmap
9	BYTE	Reserved
10	ADDR	Pointer to destination bitmap
14	WORD	Width of destination bitmap
16	WORD	Height of destination bitmap
18	P0	Coordinate of destination data
18+p	BYTE	Format of source bitmap
19+p	BYTE	Reserved
20+p	ADDR	Pointer to source bitmap
24+p	WORD	Width of source bitmap
26+p	WORD	Height of source bitmap
28+p	P1	Coordinate of source data
28+2p	BYTE	Format of pattern bitmap
29+2p	BYTE	Reserved
30+2p	ADDR	Pointer to pattern bitmap
34+2p	WORD	Width of pattern bitmap
36+2p	WORD	Height of pattern bitmap
38+2p	P2	Coordinate of pattern data

Figure 4-13. HBMC: Bitmap Copy

Description: The HBMC order copies a rectangular image block to a given position in the destination bitmap. The block copied can be:

- from the source bitmap with no pattern bitmap
- from the source bitmap under control of the pattern bitmap
- from pattern to destination with no source bitmap.

If the areas overlap for an inter-bitmap copy (that is, the source bitmap equals the destination bitmap), the destination overlays the source. Otherwise, the source bitmap remains unchanged.

The source and destination bitmap formats must be equal, and can have a value of 1, 2, 4, or 8 bits per pel. The pattern bitmap format must have a value of 1 bit per pel.

The size of the bitmaps and image block are given in the width and height parameters, in pels.

The bitmap coordinates refer to the top left of the image in order space.

The pattern bitmap format is 1 bit per pel. For a copy from source, under control of the pattern, a 0 in the pattern uses the background color and mix; a 1 in the pattern uses the foreground color and mix. For a copy of the pattern to destination, a 0 in the pattern uses the background color and mix.

Usage Notes

- An "invert bitmap" bit specifies that the bitmap is inverted during copy, that is, the line of data (y) is at:

  $$(\text{bitmap height} - y - 1)$$

  after the bitmap copy, where:

  $$0 \leq y \leq (\text{bitmap height} - 1).$$

- The pointer is a 32-bit offset from the start of the display RAM.
- The HBMC order supports "non-byte-wide" source and pattern bitmaps.
- The destination bitmap must start on a DWORD boundary, and the width of the bitmap must be an integer multiple of DWORD's wide. That is, the result of the division, maps width/pels per byte, must be an integer multiple of 2.

Attributes

- Color mix
- Color comparison.

Control by the Set Scissor function is only applied if the destination is the current bitmap. The current position is not affected.

HSDW - Set Display Window

Function: The HSDW order sets the display window within the screen bitmap in display direct access storage.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (4+p)
2	P0	Coordinate of display window origin (top left)
2+p	WORD	Window width
4+p	WORD	Window height

Figure 4-14. HSDW: Set Display Window in Screen Bitmap

Description: The HSDW order positions the display window within the screen bitmap in display direct access storage, and sets its height and width.

Usage Notes

- The X-coordinate of the display window origin does not operate on a pel resolution. The resolution is determined by the need for the display window to start on a QWORD boundary.

Extended Function Set (2)

HSPRITE - Sprite at Given Position

Function: The HSPRITE order draws the current Sprite shape at a given position.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (p)
2	P0	Coordinate data of Sprite

Figure 4-15. HSPRITE - Sprite at Given Position

Description: This order draws the Sprite shape in screen space at point P0.

Usage Notes

- The Sprite is positioned so that the "hot point" is located at the coordinate specified (P0).

  For a definition of the "hot point," see "HSSPRITE - Set Sprite Shape" on page 4-26 .
- The Sprite can only be positioned within screen space, that is, the display RAM bitmap for the screen.
- The HSPRITE order is ignored if the destination bitmap is not the screen bitmap.
- The HSPRITE order is not subject to the drawing operation attributes or controls.

HSSPRITE - Set Sprite Shape

Function: The HSSPRITE order defines the shape of the Sprite.

Entry Point Parameter Block

Byte	Content	Meaning
0	WORD	Length of following data (24 or 1)
2	BYTE	Flags: Bit 7: = 0: Visible = 1: Invisible Bits 6—0, Reserved (must be 0)
3	BYTE	Reserved (must be 0)
4	BYTE	"Hot point" x offset
5	BYTE	"Hot point" y offset
6	DWORD	Sprite image definition address
10	WORD	Sprite image width
12	WORD	Sprite image height
14	WORD	Color 1 green value
16	WORD	Color 1 red value
18	WORD	Color 1 blue value
20	WORD	Color 2 green value
22	WORD	Color 2 red value
24	WORD	Color 2 blue value

Figure 4-16. HSSPRITE - Set Sprite Shape

Description: The HSSPRITE order specifies the image of the current Sprite shape.

The Sprite image format is 2 bits per pel, and should be "packed": that is, bits 0—1 of the first byte define pel 1, bits 2—3 of the first byte define pel 2, and so on. "Padding" bits should be inserted between rows, so that a new row always starts on a byte boundary.

The meaning of the Sprite image pel values is as follows:

Bits 1—0	Sprite effect
B'00'	Color 1
B'01'	Color 2
B'10'	Transparent
B'11'	Reserved

The *hot point* is the position of the *pick point*, as used by applications, within the Sprite shape. The hot point is defined by offsets from the top left corner

of the Sprite shape, and can be positioned anywhere within screen coordinates.

The Sprite can be made invisible by setting the flag byte in the parameter block: the length field should be 1.

Usage Note: The XGA hardware Sprite can be of any size, up to 64 x 64, that is a multiple of four, and therefore does not need padding bits between rows. XGA only uses the most significant bits of the color 1 and 2 gun values.

The number of bits supported can be determined from the number of intensity levels returned by the HQMODE order.

Appendix A. Display Adapter 8514/A Compatibility

The XGA Adapter interface provides functions similar to those provided by the IBM Display Adapter 8514/A, but at an enhanced level, using what is in effect a superset of the 8514/A orders. However, there are significant hardware differences between XGA and the Display Adapter 8514/A.

Although most applications written to use the 8514/A can also be run using XGA with the XGA Adapter Interface, some will not run correctly. These include:

- **Applications Using the 4 x 4 Mode**

  In the 8514/A, with its four-plane, 512KB display buffer storage, 256 colors could be displayed by treating the buffer as two separate banks of four planes, under application control. This 4 x 4 mode is not supported by the XGA Adapter Interface.

- **Dual Display Buffer Applications**

  The 8514/A display buffer is completely separate from that used by the Video Graphics Array (VGA). Data stored in the VGA buffer—while the 8514/A was operating in Advanced Function mode—would therefore be displayed when an HCLOSE order was executed. However, the XGA video sub-system has only one buffer, and cannot therefore support a dual display buffer application unless a dual display configuration is provided.

- **Applications Using Non-Visible Display Buffer Storage**

  Both XGA and the 8514/A have areas of display storage that are not used for "screen refresh" purposes. These areas are used to store fonts and perform other housekeeping tasks. However, it is possible for an application to use this "offscreen" storage to store data temporarily. Any 8514/A application that uses the offscreen area for temporary data storage might therefore erase fonts, or other essential data, when run on XGA.

Appendix B. Default Palettes

The advanced-function default palette provides two banks of 16 colors; the palette is loaded with the 16 values shown in the figure.

Planes 4 - 7 ↓

Planes 0 - 3

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

Figure B-1. Default Palette Values

The meaning of the values depends on whether there is a color or monochrome display attached to the adapter.

For a color display, the palette values X'0'—X'F' are treated as having the bit significance I, R, G, or B. Hence, the displayed colors are as follows:

I	R	G	B	Color
0	0	0	0	Black
0	0	0	1	Blue
0	0	1	0	Green
0	0	1	1	Cyan
0	1	0	0	Red
0	1	0	1	Magenta
0	1	1	0	Brown
0	1	1	1	White
1	0	0	0	Gray
1	0	0	1	Light blue
1	0	1	0	Light green
1	0	1	1	Light cyan
1	1	0	0	Light red
1	1	0	1	Light magenta
1	1	1	0	Yellow
1	1	1	1	High intensity white

For a monochrome display, the palette values are defined so that color applications show good gray-level discrimination. The following table defines the display intensity (on a scale 0—63) for the palette values X'0'—X'F':

Palette Value	Display Intensity
0	0
1	5
2	17
3	28
4	8
5	11
6	20
7	40
8	14
9	24
A	45
B	50
C	32
D	36
E	56
F	63

Appendix C. Font File Format

Three fonts are supplied with the adapter interface installation code; each font contains five language code pages, as follows:

Code Page ID	Language
437	US/English
850	Multilingual
860	Portuguese
863	Canadian/French
865	Nordic

The adapter interface code installation procedure allows both a default (for example, 437—US English) and an alternative default (for example, 850—Multilingual) code page ID to be selected for applications which use these supplied fonts. A code page ID (for example, 437) is an ASCII string followed by a null.

Each code page has a *Character Set Definition Block* (see "Programmable Character Set Specification" on page 3-70). The five address fields within the character set definition block are offsets within the file; the address at which the font is loaded *must be added* to give the correct values.

Each font file contains a header (see the table below) that is updated by the installation procedure. Offsets given in the table are within the file to the Character Set Definition Block (CSDB) of the code page specified.

Byte	Content	Meaning
0	WORD	Number of code pages within this file
2	WORD	Default code page index into table below (range 0 to 4)
4	WORD	Alternative default code page index into table below (range 0 to 4)
6	STRING	4-byte ID (437) of first code page in file
10	WORD	Offset to CSDB of first code page
12	STRING	4-byte ID (850) of second code page in file
16	WORD	Offset to CSDB of second code page
18	STRING	4-byte ID (860) of third code page in file
22	WORD	Offset to CSDB of third code page
24	STRING	4-byte ID (863) of fourth code page in file
28	WORD	Offset to CSDB of fourth code page
30	STRING	4-byte ID (865) of fifth code page in file
34	WORD	Offset to CSDB of fifth code page

Index

D

E

F

G

H

I

L

M

O

P

R

S

T

W

Printed in the
United States of America