GRAPHICS/1000-II
Advanced Graphics Package - 3D
User's Guide

HEWLETT
PACKARD

HEWLETT-PACKARD COMPANY
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014

MANUAL PART NO. 92842-90002
Printed in U.S.A. March 1982

PRINTING HISTORY

The Printing History below identifies the Edition of this Manual and any Updates that are included. Periodically, Update packages are distributed which contain replacement pages to be merged into the manual, including an updated copy of this Printing History page. Also, the update may contain write-in instructions.

Each reprinting of this manual will incorporate all past Updates, however, no new information will be added. Thus, the reprinted copy will be identical in content to prior printings of the same edition with its user-inserted update information. New editions of this manual will contain new information, as well as all Updates.

To determine what manual edition and update is compatible with your current software revision code, refer to the appropriate Software Numbering Catalog, Software Product Catalog, or Diagnostic Configurator Manual.

Second Edition .... Mar 1982

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Preface

Who needs to use this manual?

The AGP-3 User's Guide is intended for users of Hewlett-Packard's Advanced Graphics Package - 3D (AGP-3). Graphics application designers will appreciate it as a general overview of AGP-3's capabilities. Application programmers will use it as they write their first graphics programs, enjoying its conceptual descriptions and programming examples. Operators of graphics applications who want to better understand the system they use may read the first two chapters of this manual.

What does it cover?

The AGP-3 User's Guide provides a tutorial introduction to AGP-3. It teaches the new user by introducing basic graphics concepts and AGP-3's implementation of them. This manual does not attempt to describe all details of the AGP-3 system, as this would impair the learning process.

Two other manuals, the AGP-3 Reference Manual and the GRAPHICS/1000-II Device Handler's Manual, will be of interest to AGP-3 users. The Reference Manual provides additional product documentation. As more sophisticated applications are addressed, it will be relied on for complete descriptions beyond the introductory level of the User's Guide. The Device Handler's Manual has a different purpose. It documents details of the interaction between AGP-3 and the graphics devices being used.

What does it assume?

Very little is assumed by the AGP-3 User's Guide. An overall understanding of basic concepts of computers (e.g., programs, peripherals, etc.) is necessary. Some familiarity with computer graphics is helpful, but a thorough reading and understanding of Chapter 2 will provide it. Also, knowledge of the FORTRAN programming language is desirable, since that is the language in which examples are written. Experience with additional host languages, such as Pascal or Assembler, would complement the FORTRAN knowledge and allow those languages to be used for graphics production.

How is it organized?

This manual is organized as a tutorial. To benefit from the tutorial approach, the new user should read the chapters in the order in which they are presented.

The chapters are:

Chapter 1	An overview of the AGP-3 product. Graphics production, features, and host language considerations are covered in this chapter.
Chapter 2	General concepts of computer graphics. An overview of AGP-3's five functional areas is presented. The remaining chapters treat these areas in detail.
Chapter 3	Output Primitives. The use and attributes of the fundamental elements of a graphics picture is the subject of this chapter.
Chapter 4	Viewing Transformations. The process of producing graphics images from two or three-dimensional objects is explained.
Chapter 5	Segments. The ability to subdivide complex pictures into separate logical entities is discussed.
Chapter 6	Input. The steps for receiving graphics input from different types of devices are presented.
Chapter 7	Control. The remaining functions, necessary to control the graphics environment, are the subject of the final chapter.
Appendix A	Errors. A complete list of AGP-3 errors, ordered numerically, is found in this appendix.
Appendix B	Pascal Considerations. Specific recommendations and constraints when using AGP-3 from application programs written in Pascal are covered in this appendix.
Glossary	General computer graphics terms and AGP-3's specific use of them are documented in the glossary.

On the cover: Albrecht Durer, "Artist Drawing a Lute"
(The Metropolitan Museum of Art, Harris Brisbane Dick Fund, 1941).

Table of Contents

List of Illustrations

List of Tables

Chapter 1
Overview

General

A computer graphics package is a tool to help computer users utilize graphics devices. Graphics devices are hardware peripherals that draw lines and characters in response to commands from an application program, producing pictures based on the information they are sent. A computer graphics package is a software layer between the application program and the graphics device that makes the application program easier to write and maintain.

Advanced Graphics Package - 3D (AGP-3) is a computer graphics package. AGP-3 provides the application program with simplified access to the devices necessary to produce computer graphics, assuming much of the overhead associated with interactive graphics creation. It relieves the program of the burden of using commands and units specific to a device. This idea, known as <u>device independence</u>, means that the internal logic of the application program may remain unaffected by the type of graphics device being used. Device independence is desirable for several reasons. It makes the application program easier to write initially, easier to maintain over time, easier to enhance through the addition of new peripherals, and easier to transport from one system to another. With these features, AGP-3 makes the advantages of computer graphics available while simplifying its utilization.

Division of the Graphics Task

With AGP-3, application programs describe <u>objects</u> in a graphical world. Objects are specified in the world coordinate system, a three-dimensional Cartesian coordinate system. The world coordinate system's units are the units of the application; that is, the object can be described in any units that the application requires.

In general, objects are three-dimensional, though the nature of some applications may make use of the third dimension unnecessary. These applications may choose to ignore the depth of objects by implicitly working in the X-Y plane.

AGP-3 produces images from objects. The image of an object is the flat representation of it on a graphics display device. Television produces images on a television set's screen, while AGP-3 produces images on a display device.

Images are produced from objects in two steps, using the virtual coordinate system. Virtual coordinates are two-dimensional and range between 0.0 and 1.0. They allow an object to be translated into units that are neither application-dependent nor device-dependent.

As shown in Figure 1-1, the two steps in the production of images from objects are:

1. Application data represented in the world coordinate system is translated to virtual coordinates by projection from three dimensions to two dimensions.

2. Virtual coordinates are translated to the units and commands needed by individual display devices.

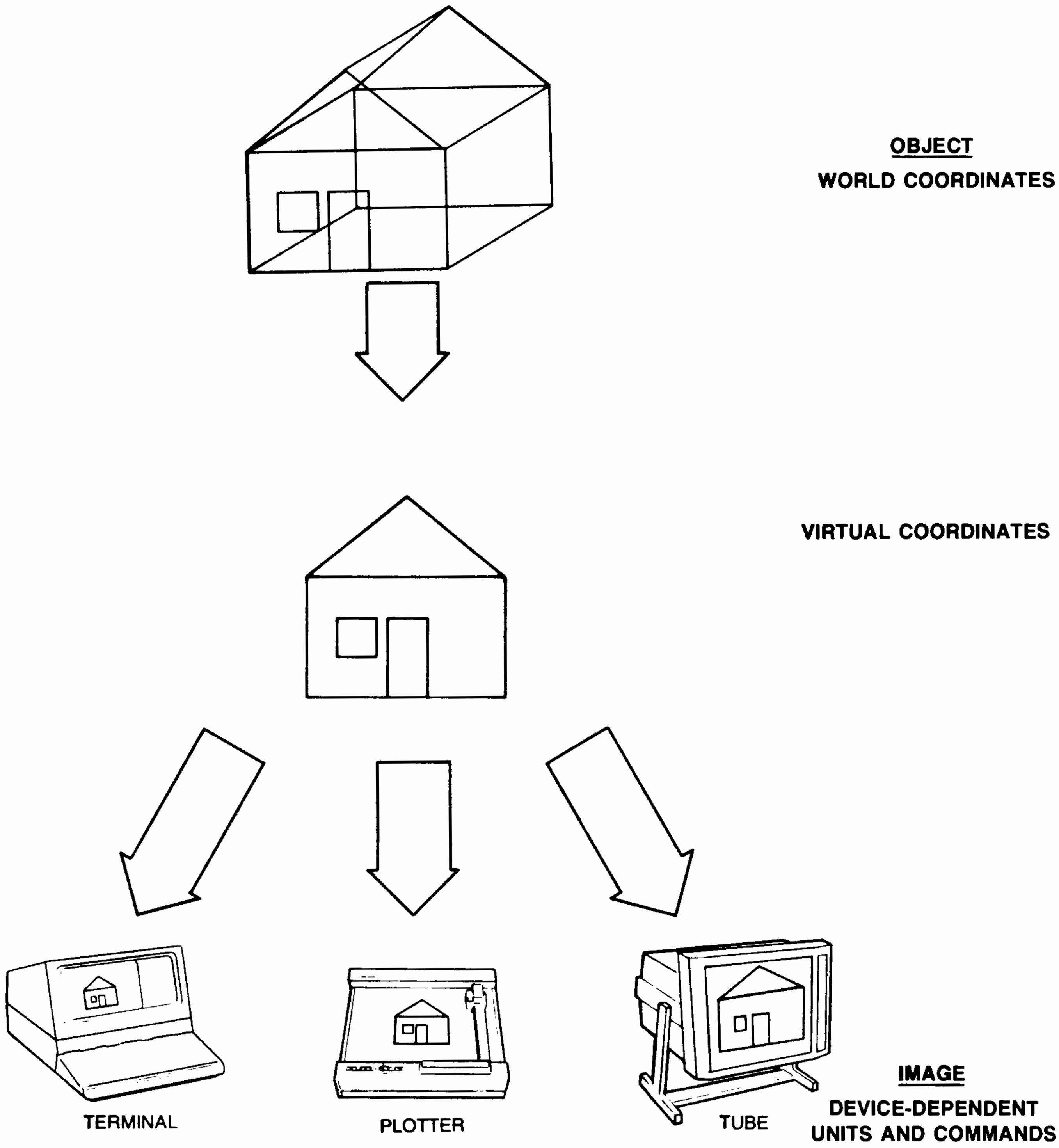

Figure 1-1. Images Are Produced From Objects.

AGP-3 is broken into two separate programs to reflect this division of processes. Each is loaded individually and runs on the host operating system in a different memory partition. Together they provide a high-level interface to the graphics devices being used.

User Program

Generating computer graphics with AGP-3 requires writing a user program (UP). The user program is an application program containing calls to AGP-3 routines. The parameters to these routines control the graphics production process, allowing information to be passed both to and from AGP-3. Routines referenced by the user program are appended to it by searching the library $UPLIB (User Program LIBrary) when the user program is loaded.

Data in the user program is represented in terms of the world coordinate system. The AGP-3 routines appended to the user program generate virtual coordinate data from the user's world coordinate data. This virtual coordinate data is then transmitted to a second program, called the work station program, through Class I/O. Class I/O is a feature of the RTE operating system that can be used for program-to-program communication.

Work Station Program

Each work station program (WSP) controls a graphics work station. A work station is a collection of devices that are treated as an identifiable unit. Every work station has its own work station program. Unlike the user program, the work station program is not written by the user, but is configured and loaded with standard parts provided in AGP-3.

Each work station program knows how to translate device-independent virtual coordinate data from a user program into device-dependent commands for a particular device. A work station may include the following logical devices (a logical device is the abstraction of a physical device):

1. graphics display
2. alphanumeric display
3. button input
4. keyboard input
5. locator input
6. valuator input
7. pick input

Two types of output devices are available: graphics and alphanumeric. Of the input devices, a button device returns an integer, a keyboard device returns alphanumeric text, a locator returns a real coordinate pair, a valuator returns a single real value and a pick device identifies a displayed image. A work station may have one device of each type.

The portion of a work station program that converts virtual coordinate data to device-dependent commands is known as a device handler. A device handler controls a single device on a work station; for example, one is used for the HP 2648A Graphics Terminal's keyboard. A second device handler would be necessary for its screen. The device handlers in the work station program are part of DGL (Device-independent Graphics Library). AGP-3 relies on DGL for this service.

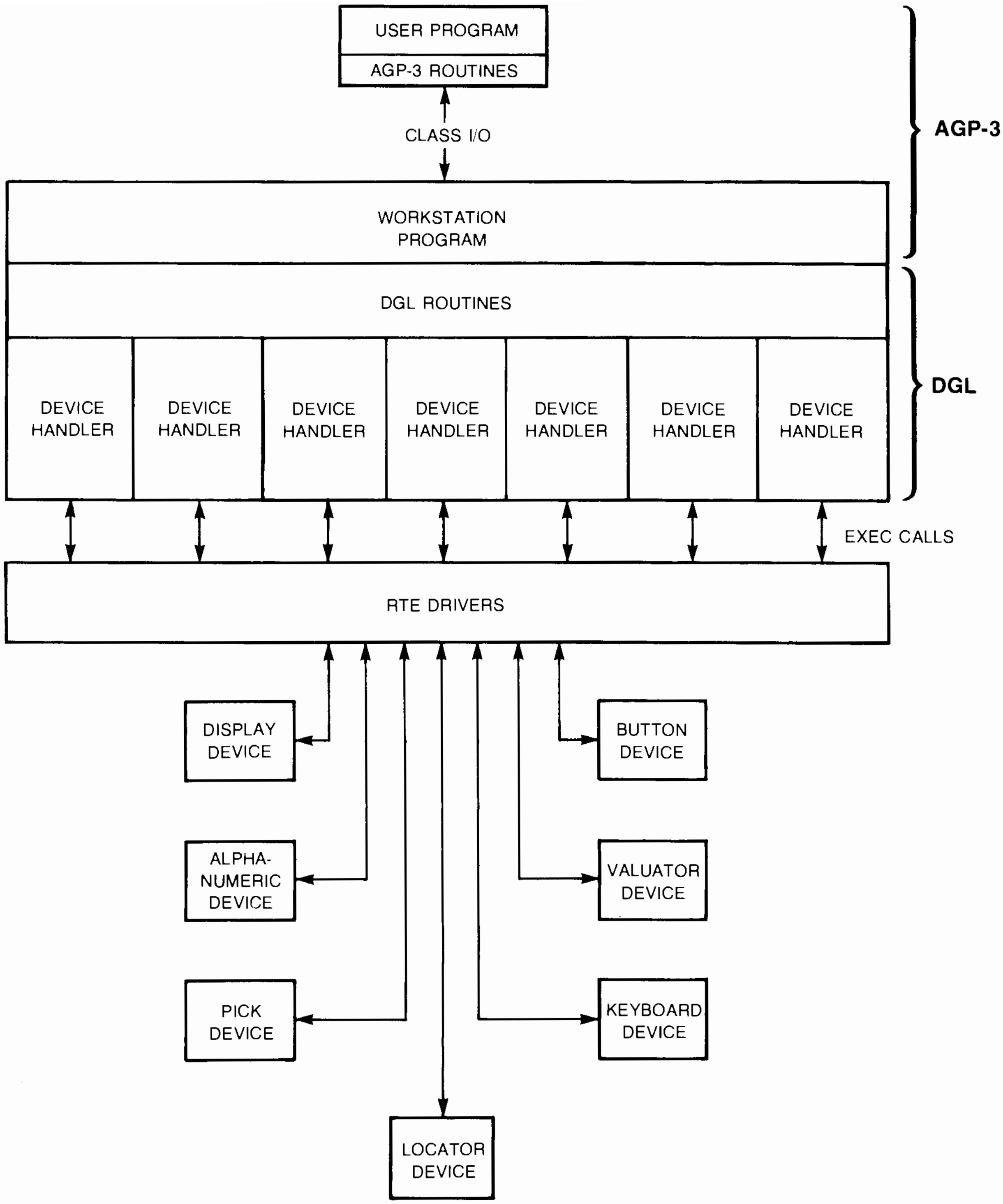

Figure 1-2. Relationship Between AGP-3 and DGL.

Throughout this manual, certain features will be described as "device-dependent", meaning they are a function of the device handlers provided by DGL and of capabilities of the particular devices being used. The GRAPHICS/1000-II Device Handler's Manual should be referred to in these cases. For example, the devices supported by DGL are exactly those devices that AGP-3 may use. The Device Handler's Manual provides a list of these devices, as well as device-dependent information affecting AGP-3.

Advantages of the Division

The graphics system is divided into two separate programs for a number of reasons. First, the user program is device-independent. Once written, it can control any supported graphics device by choosing the appropriate work station program when it is executed.

A second advantage is that the user program can control more than one work station concurrently. The same output is directed to each without having to duplicate portions of user program code. For example, both a terminal version of a graphics image and a hard copy on a plotter could be produced simultaneously. Two work stations would be specified before the first image was output; as many as eight could be chosen by initializing the appropriate work station programs.

Since much of the AGP-3 code necessary to perform the graphics task is in the work station program, less of the user program's address space is needed for the AGP-3 support routines. Therefore, another advantage of the two program approach is that more logical address space is available to the user.

Finally, loading procedures are simplified and the time to load a graphics program is reduced by dividing the graphics task. Standard work station programs need only be loaded once when AGP-3 is installed in the system. User programs are loaded more frequently, but since they contain only a part of the code necessary to produce graphics, the loader will have less relocation to do and so will finish more quickly.

Host Languages

AGP-3 routines may be called from user programs written in FORTRAN IV, FORTRAN 4X, FORTRAN 77, Pascal or Assembler languages.

FORTRAN

Application programs written in FORTRAN IV, FORTRAN 4X, or FORTRAN 77 are entirely compatible with AGP-3. Parameters passed to AGP-3 routines should be either one-word integers or two-word real numbers.

The programming examples of this manual are written in FORTRAN. They are not complete programs, however. Care should be exercised when combining them into your own programs, because many calls that must be used (but are not related to the topic being presented) are not shown. For example, the initialization and termination routines are not described until the last chapter of this manual, but they are needed in every program.

The following FORTRAN example is complete. In addition to its value as an example that may be typed into the computer and executed, it may be modified to include many of the program fragments that are listed in later chapters.

```
FTN4,L
      PROGRAM USER
C
C..AGP-3 user program to draw a line.
C  The work station program name is hard coded to 'WSP'.
C  The LU for graphics output is hard-coded to 1.
C
C..Initialize the AGP-3 system.
C
      CALL JBEGN
C
C..Initialize a work station program:
C  work station 1, name='WSP', graphics output LU=1,
C  no control bits set. Then enable it for graphics output.
C
      CALL JDINT (1, 6HWSP   ,1,0)
      CALL JWON  (1)
C
C
C..Everything between the comment lines may be changed as needed.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C..Use default attributes and viewing transformation to draw line.
C
      CALL J2MOV (-1.0,-1.0)
      CALL J2DRW (1.0,1.0)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C..Disable the workstation, terminate it, end the AGP-3 system,
C  and terminate this program.
C
      CALL JWOFF (1)
      CALL JWEND (1)
      CALL JEND
      END
```

Assembler

Application programs written in Assembler are compatible with AGP-3. To call AGP-3 routines, refer to the RTE Assembler Manual section on calling FORTRAN subroutines. An Assembler version of USER, which was listed previously in FORTRAN, is given below.

```
ASMB,L
      NAM USER     AGP-3 user program to draw a line
      EXT JBEGN,JDINT,JWON,J2MOV,J2DRW,JWOFF,JWEND,JEND,EXEC
*
*..Constants:
*
ID    DEC 1        work station id
WSP   ASC 3,WSP    work station program name
GLU   DEC 1        graphics output lu
CNTRL DEC 0        initialization control word
M1.0  DEC -1.0     x and y of the beginning of the line
D1.0  DEC 1.0      x and y of the line's endpoint
D6    DEC 6        exec code for program termination
*
*
USER  NOP
      JSB JBEGN    initialize the AGP-3 system
       DEF *+1       return address
*
      JSB JDINT    initialize a work station program
       DEF *+5
       DEF ID        work station 1
       DEF WSP       name = 'WSP'
       DEF GLU       output lu = 1
       DEF CNTRL     no control bits set
*
      JSB JWON     turn it on for graphics output
       DEF *+2
       DEF ID        work station 1
```

```
*
*..Everything between the stars may be changed as needed.
*******************************************************************
*
*..Using default attributes and viewing transformation, draw a line
*
      JSB J2MOV    move absolute
       DEF *+3
       DEF M1.0      x = -1.0
       DEF M1.0      y = -1.0
*
      JSB J2DRW    draw absolute
       DEF *+3
       DEF D1.0      x = 1.0
       DEF D1.0      y = 1.0
*
*******************************************************************
*
      JSB JWOFF    disable the work station
       DEF *+2
       DEF ID
*
      JSB JWEND    terminate it
       DEF *+2
       DEF ID
*
      JSB JEND     end the AGP-3 system
       DEF *+1
*
      JSB EXEC     terminate this program
       DEF *+2
       DEF D6
      END USER
```

Pascal

Due to differences between FORTRAN and Pascal, care must be taken when writing application programs in Pascal. In general, the difficulties that arise when accessing AGP-3 routines are those that arise when accessing <u>any</u> non-Pascal routine from a Pascal program. Therefore, a review of that topic in the Pascal/1000 Reference Manual may be useful. Additionally, Appendix B provides a detailed list of considerations and suggestions for the application programmer who uses Pascal.

All AGP-3 routines referenced in a Pascal program should be declared as external procedures. Also, their actual parameters should be of an appropriate data type. Files are supplied with the AGP-3 product that provide standard data types and external declarations for inclusion in all application programs. The contents of these files may be modified as needed.

A Pascal version of program USER, which was listed previously in FORTRAN and Assembler, is given below.

```
PROGRAM user;    { AGP-3 user program to draw a line.              }

{ The following three include files are part of the AGP-3 product.}
$INCLUDE '[PAGP1'                { constant and type declarations }
$INCLUDE '[PAGP2'                { external procedure declarations}
$INCLUDE '[PAGP3'                { more external procedure decls. }


BEGIN  { Main of user }

{ Initialize the AGP-3 system.                                     }
  jbegn;

{ Initialize a work station program:                               }
{ work station 1, name='WSP', output LU = 1, no control bits set. }
{ Then enable it for graphics output.                              }
  jdint (1,'WSP   ',1,0);
  jwon (1);

{ Everything between the dollar signs may be changed as needed.   }
{ $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ }

{ Using default attributes and viewing transformation, draw a line}
  j2mov (-1.0,-1.0);             { move absolute }
  j2drw ( 1.0, 1.0);             { draw absolute }

{ $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ }

{ Disable the work station, terminate it, and end the AGP-3 system}
  jwoff (1);
  jwend (1);
  jend;
END.  { Main of User}
```

Chapter 2
Concepts of Computer Graphics

Introduction

Computer graphics is an important step toward utilizing computers as a communications tool, rather than just a computation device. Through graphics, information is transmitted in picture form, instead of as alphanumeric symbols whose meaning may not be immediately apparent. The graphical picture can be understood at a glance, saving the viewer from tediously analyzing tabular data. In this way, ideas rather than letters and numbers can be communicated quickly and efficiently, increasing the productivity of the communication process.

The following sections describe five major functional areas of AGP-3, which would be present in one form or another in almost any high level graphics product. The concepts and examples introduced here will be developed in detail in the following chapters.

Graphics Output Primitives

Graphics output primitives are the building blocks of objects. Just as an algorithm must be broken down into the simplest possible instructions to create a computer program, a graphically created object must be reduced to output primitives. AGP-3 uses five types of output primitive: moves, lines, polylines, text, and markers. Lines and text are familiar concepts. A move is just that, a move from one location to another. A polyline is a sequence of connected lines, and markers are data symbols, such as diamonds or asterisks. In Figure 2-1, markers indicate data points on a graph, while a series of line primitives are used to connect the data points.

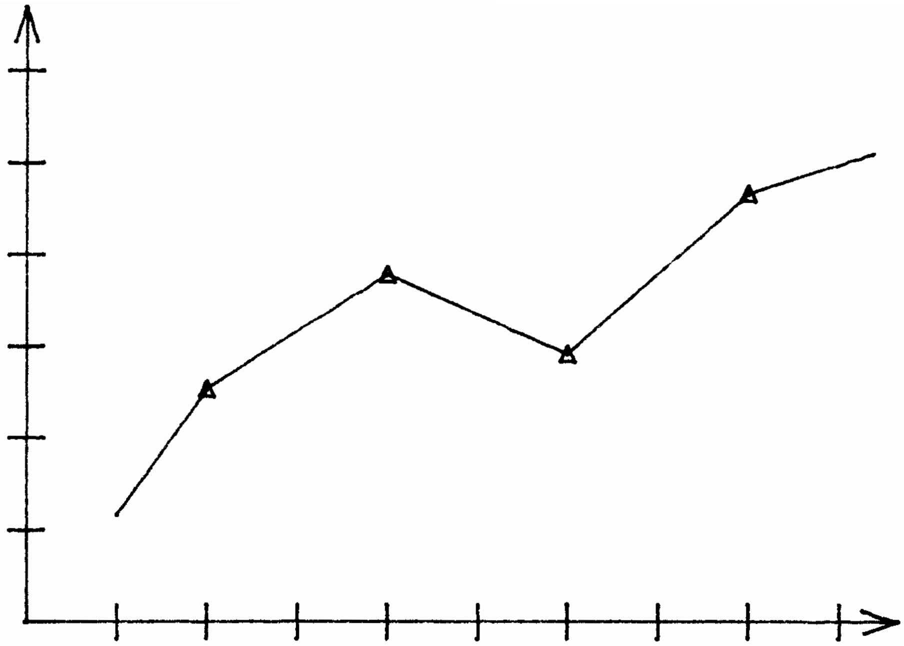

Figure 2-1. Output Primitive Examples.

The location of each output primitive is specified in the world coordinate system, with a point denoted by its X, Y, and Z coordinates. The graphics software "remembers" the most recently specified point, known as the current position. A line can then be created by specifying another point, which AGP-3 connects to the first. This is known as a "draw." The end of that line then becomes the current position, and another line can begin there. The current position can also be moved to another point without drawing a line; this is known as a "move." A picture can be drawn by successive moves and draws.

The polyline primitive makes it possible to draw a series of connected lines all at one time, instead of making a series of draws. This is done by specifying the number of points to be drawn and the X, Y, and Z arrays of the coordinates to be connected.

Text can be of high, medium, or low quality. High quality text is software-generated from a series of line primitives and can appear in a variety of different fonts. Medium and low quality text are usually generated in hardware. Since display devices typically cannot produce all sizes and orientations of text, the appearance of medium and low quality text conforms less fully to the requirements of an application program than high quality text does.

Each character in a medium quality text string is placed individually, while a low quality string is placed as a unit. Thus only the starting point of a low quality text string is under the application program's control (see Figure 2-2).

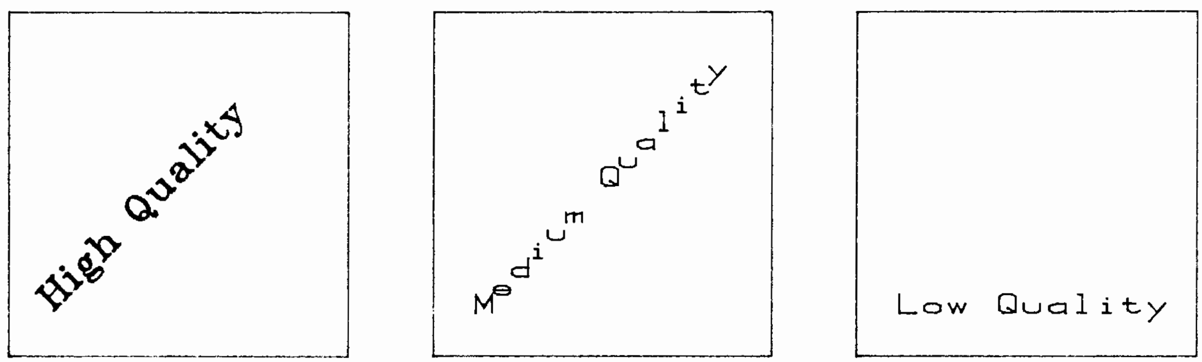

Figure 2-2. High, Medium, and Low Quality Text.

With the output primitives of line, polyline, text, and marker, and the ability to move the current position, complex graphical pictures can be constructed. Figure 2-3 shows the Gossamer Condor, a particularly interesting graphics figure consisting of nothing more than line primitives.

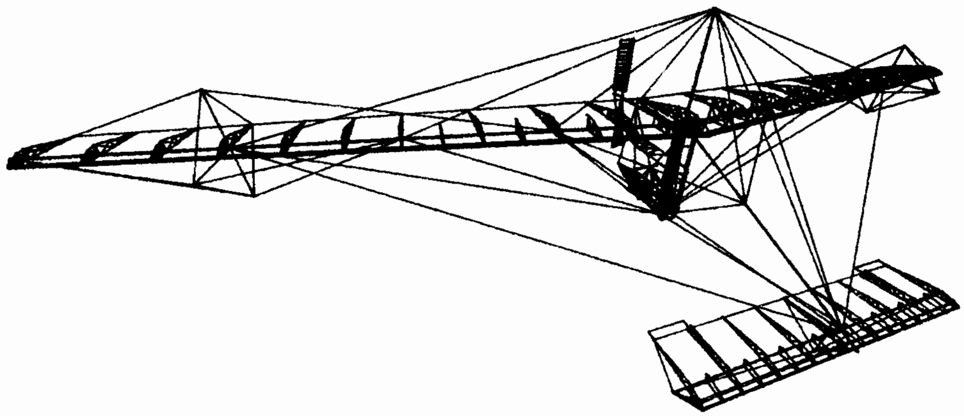

Figure 2-3. Complex Figures Can Be Drawn.
(Courtesy of Dr. Paul MacCready, president of AeroVironment, Inc., designer of the Gossamer Condor. Data created by Paul Lionikis and Prof. Charles L. Owen, Institute of Design, Illinois Institute of Technology.)

Attributes

Attributes are the general characteristics of graphical pictures. Output primitive attributes affect the appearance of individual primitives. For example, the attributes of a line primitive are color, linewidth, and linestyle. Different values of the linestyle attribute might produce the various styles shown in Figure 2-4.

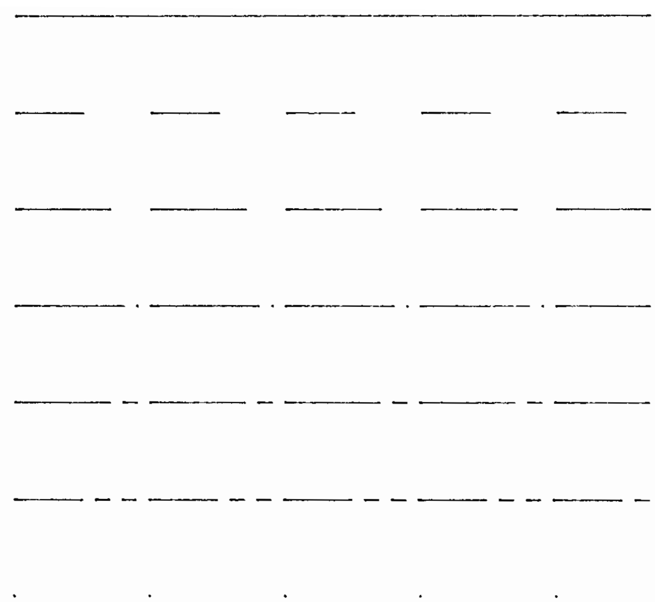

Figure 2-4. An Output Primitive Attribute: Linestyle.

Text appearance can also be controlled with primitive attributes. Low quality text has the attributes of size, justification, color, linewidth, and linestyle. Medium quality text has these attributes as well as gap and orientation. High quality text has all of these, as well as font and slant. Figure 2-5 shows examples of Eurostyle, Simplex Roman, Triplex Roman, Script, Mathematical, and Gothic text fonts.

Eurostyle

Simplex Roman

Triplex Roman

Script

μαΤηεμαΤιψαω

Gothic

Figure 2-5. Six Fonts of High Quality Text.

Viewing Transformations

As objects are created, they are transformed for viewing. A viewing transformation converts two or three-dimensional objects into two-dimensional images. The direction and distance from which an object is seen, the portion of it to be displayed, and the location of the object's image on the display device are all chosen by the user.

Renaissance artists practiced this act of transforming objects for viewing, calling it "perspective". Though television is now commonplace, the first television viewers were astounded to see their world displayed on a flat screen. Today, a camera accomplishes in a moment what took an artist years to master: portraying three dimensions in two.

AGP-3 is capable of both two and three-dimensional viewing transformations. Two-dimensional viewing is a subset of three-dimensional viewing, where the Z-axis of the world coordinate system is ignored. Since the two-dimensional transformation is the simpler case, it is discussed first.

Two Dimensions

In defining the two-dimensional viewing transformation, three questions need to be addressed:

1. What portion of the world coordinate system will be displayed (where is the window)?
2. Where will the window be mapped in the virtual coordinate system (where is the viewport)?
3. Where will the virtual coordinate system be placed on the display device (what are the logical display limits)?

Setting the window can be thought of as taking a large sheet of paper and cutting a rectangular hole in it, then placing the piece of paper over the object of interest. Only what is inside the hole, or window, will be visible. In AGP-3, the window is specified as a rectangular area of the world coordinate system. Any portion of an output primitive lying inside the window is displayable; any portion outside is not. In some cases, the window may be large enough to see an entire object; in others, it may be more desirable to see part of the object while not showing the remainder. Figure 2-6 shows the effect of two different windows on the same object.

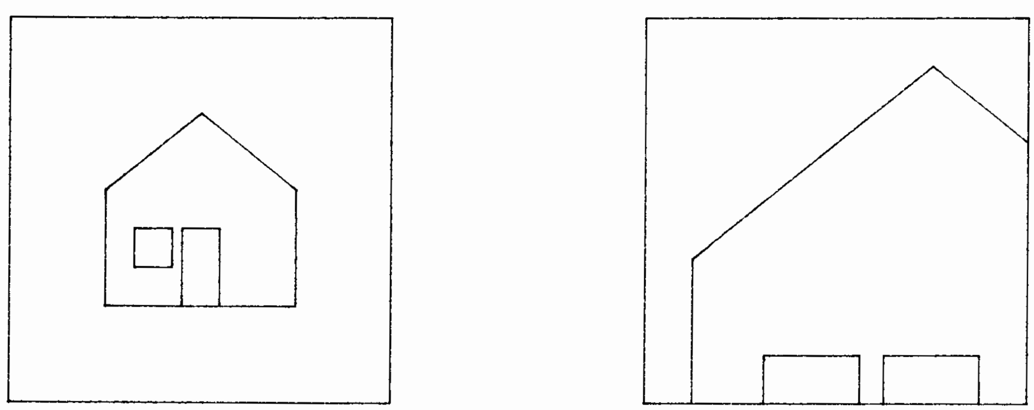

Figure 2-6. Two Windows On the Same Object.

The area on a view surface where images appear is called a viewport. The dimensions of the viewport are set in the virtual coordinate system, which is the two-dimensional system whose units range from 0.0 to 1.0. Frequently, the entire view surface will be the viewport, or multiple viewports can be defined sequentially so that several images can be displayed together. Figure 2-7 shows such an example. On a HP 2648A Graphics Terminal a left and right viewport have been set, allowing one image to be displayed on the left and another on the right.

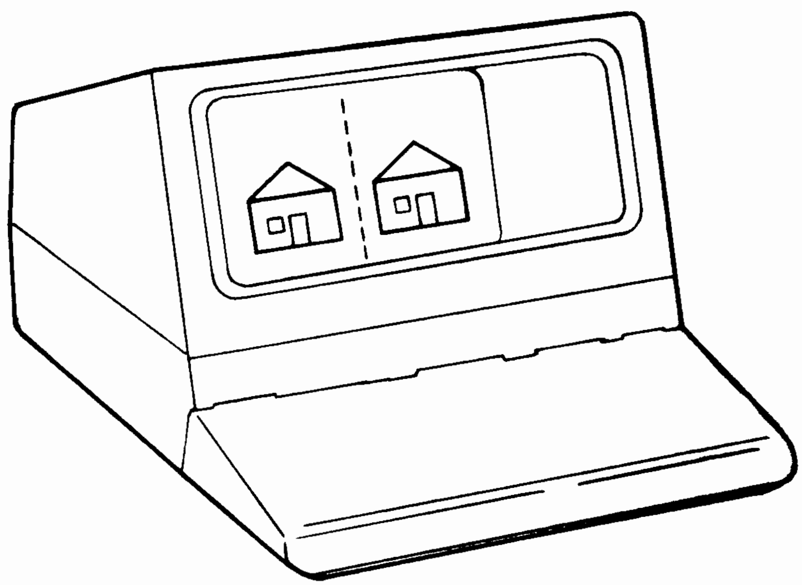

Figure 2-7. Two Viewports, One Object.

Once the window and viewport have been defined, output primitives within the window are mapped into the viewport so that each point in the window corresponds to a point in the viewport.

To offer full flexibility in accessing display devices of different sizes and shapes, AGP-3 lets the application program control the mapping of the virtual coordinate system to the display device. The logical display limits can be manipulated to alter the absolute size of the picture that is produced. These limits bound the <u>logical display surface</u>, perhaps making it the entire display device, or often defining some subset of it. The adjective "logical" distinguishes this surface from the surface of the device itself, sometimes called the physical surface.

The steps in the mapping of objects in the window to images on a display device's logical display surface are somewhat complex. They are described more fully in Chapter 4, where the AGP-3 routines that control this mapping are presented.

Three Dimensions

Three-dimensional viewing is more complex than two-dimensional viewing because it can be done from more than one direction. Three-dimensional viewing transformations are very powerful in nature. With relatively little application code, displayed images can be given a variety of perspectives and appearances.

AGP-3 mathematically determines the appearance of three-dimensional objects by transforming the application program's data. A representation of this process is shown in Figure 2-8, a medieval woodcut by Albrecht Durer. Using a mathematical version of Durer's mechanical approach, AGP-3 can create various views of an object. By altering the viewing transformation, the best view for a particular application can be achieved.

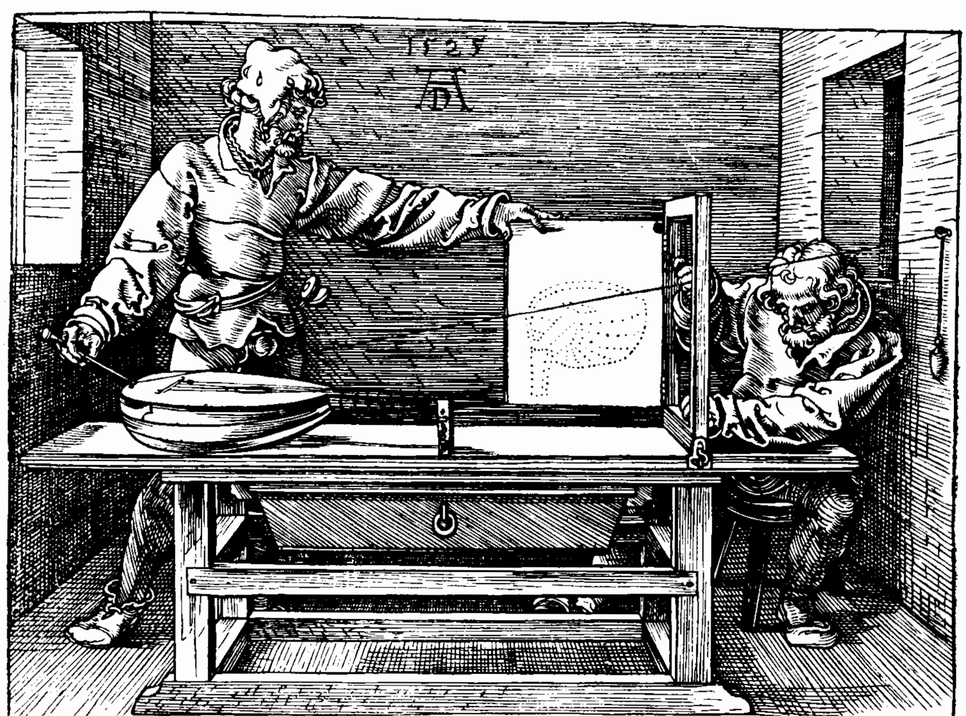

Figure 2-8. Albrecht Durer, "Artist Drawing a Lute".
(The Metropolitan Museum of Art, Harris Brisbane Dick Fund, 1941)

Durer's artist is using string to mechanically project a lute (object) through a wooden frame to a hook on the wall. The pattern formed by the strings where they intersect the plane of the frame is the transformed view (image) of the lute. This transformation of an object into an image corresponds nicely to the constructs in AGP-3 that accomplish the same thing.

AGP-3 uses a viewplane, a plane on which a three-dimensional object is reduced to a two-dimensional image by passing lines called projectors, one through each point of the object, to a single point called the center of projection. In the Durer woodcut, the wooden frame contains the viewplane, the string is a projector, and the hook on the wall is the center of projection. When the center of projection is a finite distance from the viewplane, a perspective projection is obtained. When it is at infinity, that is, when the projectors are all parallel, a parallel projection is obtained (see Figure 2-9). Since the hook on the wall is not more than a few feet from the wooden frame, Durer is portraying a perspective projection.

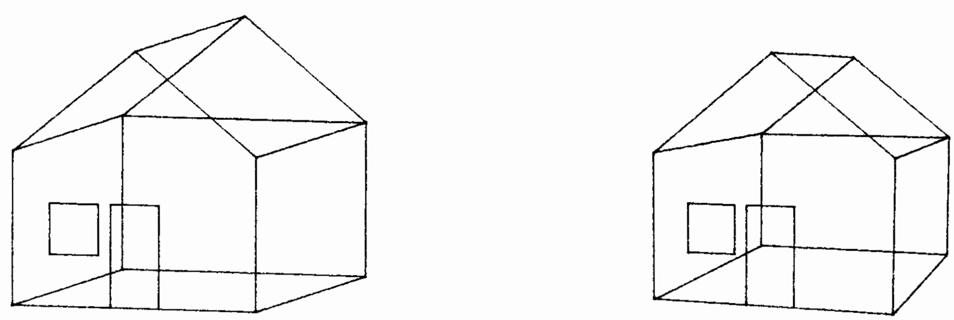

Figure 2-9. Parallel and Perspective Projections of a House.

In AGP-3, further control of an image's appearance is provided by depth clipping. Depth clipping controls how much of the object will be projected onto the viewplane. Objects in front of the hither plane and beyond the yon plane will not be projected. In Figure 2-10, the depth of a house has been clipped.

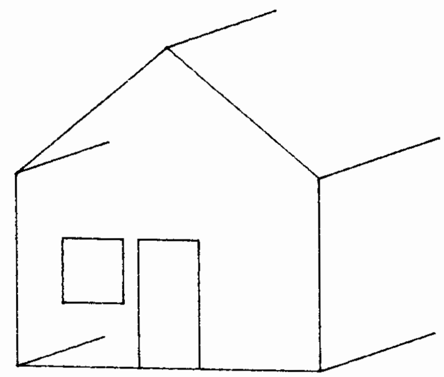

Figure 2-10. Depth Clipping of an Object.

Once the portion of the object between the depth clipping planes is projected onto the viewplane, its image has only two dimensions, so the remaining steps are similar to those of the two-dimensional viewing transformation: the projected primitives are clipped to the window, the window is mapped to the viewport, and the viewport is displayed on the display device.

Segments

A segment is a collection of output primitives that define an image. This image can be compared to a photograph, consisting of a specific object looked at from a particular viewpoint. In other words, a segment is a collection of output primitives to which a viewing transformation has been applied. One or more segments may appear on a display device simultaneously.

Segment attributes affect the appearance of segments. The segment attributes are visibility, highlighting, and detectability.

A fundamental difference between the attributes of segments and those of output primitives is their susceptibility to modification. Output primitive attributes are static, while segment attributes are not. The attributes of a primitive cannot be changed after a primitive is created. To modify a primitive's image, first you must either clear the display device or, if the primitive is contained in a segment, purge the segment. Then the primitive must be created again with the attributes in question modified.

A graphical picture can be altered easily by changing the attributes of one or more of its segments or by purging some of its segments. Using multiple segments in a picture allows changes to be made to one image without affecting those remaining, meaning a picture need not be erased and entirely reconstructed by the user in order to modify its appearance.

Output primitives are grouped into segments so the computer has an "internal" representation of the currently displayed graphical picture. This is useful for a number of reasons:

1. Primitives of a displayed image may be picked (chosen interactively) by an operator. Since segments are named, the picked image is identified by its segment name.

2. When a change is made to a picture, the display device may be updated by AGP-3 without the application program recreating all the primitives.

3. Through segments, selected parts of a picture may be erased.

4. Segments may be highlighted (to attract the viewer's attention), made invisible (to temporarily remove them from view), or made detectable (so they may be picked).

The following example shows how, in Figure 2-11, the frame of a house, its door and window, and its floor are grouped into segments so that their visibility can be manipulated independently.

```
open segment 1
  draw frame of house
close segment 1

open segment 2
  draw door and window
close segment 2

open segment 3
  draw floor
close segment 3

make segment 1 visible
make segment 2 visibile
make segment 2 invisible
make segment 3 visible
make segment 2 visible
```

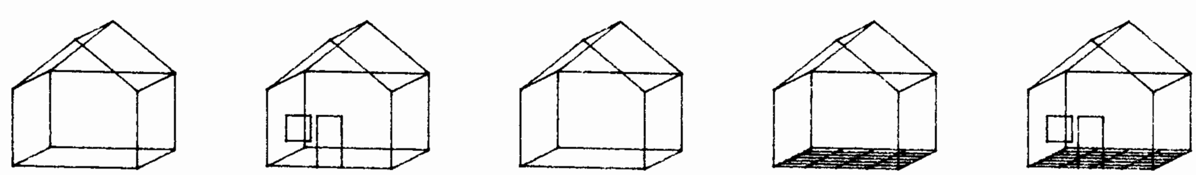

Figure 2-11. A House Made Up of Segments.

Output primitives do not have to be grouped into segments. However, the computer has no internal representation of primitives outside of segments. They are therefore temporary in nature, in that when the display is updated, those primitives will not be carried over to the next version. The removal of all primitives not contained in visible segments is called a new-frame-action. Some applications do not require all primitives to be saved, so they need not use segments.

Input

Input devices allow an operator to return information to an application program. There are five available input devices: button, keyboard, locator, valuator, and pick. A button device returns an integer value, good for applications which require only simple choices by the operator (rather than complex strings of characters such as those entered with a keyboard). A button device is analogous to a start/stop/select device, such as a button on a slide projector.

A keyboard device allows the operator to enter a string of characters, typically from an alphanumeric keyboard. The keyboard function returns to the calling program the string that was entered and the number of characters in the string. This device is useful for complex responses by operators.

A locator device provides a real coordinate pair depicting a point on the view surface. Typical locators are graphics tablets and cross-hair cursors. A possible use of this device would be designating the longitude and latitude of a city on a map.

A valuator device allows the operator to send a scalar value between 0.0 and 1.0 to the application program. The device can be a control dial or multiposition switch, analogous to a dimmer switch on a light. The dimmer can be rotated to any position, which sets the light at the desired intensity.

A pick device lets the operator select output primitives using a graphics tablet or by positioning a cursor on the desired primitives' image. The pick-ID of the primitives (assigned through AGP-3 when they were created) and the name of the segment containing the primitive(s) are returned to the application program. Using a pick device can be compared to a child pointing at the candy bar he wants in a candy store. He doesn't need to know the name of the candy. He simply sees it and identifies it by pointing. The person behind the counter, like AGP-3, identifies the candy's name for the child.

AGP-3 obtains information from the input devices by either requesting or sampling the input. Before requested information can be returned to the application program, the operator must enter a termination command (such as pressing the return key). Sampled input is returned immediately, with no operator interaction required. Information from all input devices can be obtained by the request method, while only the locator and valuator devices may be sampled.

Control

Control functions are used to manage various aspects of a graphics application. Some of these are: initialization and modification of the graphics environment, inquiry of the features currently in effect, management of error routines, and sending device-dependent escape functions. Information concerning these topics is presented in Chapter 7 of this manual.

Chapter 3
Output Primitives and Their Attributes

General

The application programmer describes objects in a three-dimensional graphical world to AGP-3 for display. These objects are comprised of five types of output primitive: move, line, polyline, marker, and text. The appearance of the output primitives result from:

1. the parameters passed to the output primitive routine (for example, the coordinates of a line, the characters of a text string, the type of marker, etc.),

2. the current values of the primitive attributes, and

3. the viewing transformation in effect.

When output primitives are created, these aspects are fixed for these primitives.

All output primitives are described in the world coordinate system using the current position as a reference point. Initially, the current position is the world coordinate origin (0.0, 0.0, 0.0). Each output primitive call results in a new current position being established. Certain operations, such as specifying a new viewing transformation or opening or closing a segment, reset the current position back to the origin.

Coordinate positions can be specified as either absolute or relative with respect to the current position. Relative coordinates are provided as a notational convenience for the programmer; AGP-3 converts them to absolute coordinates when the call is executed.

Some output primitive calls are available in either two-dimensional or three-dimensional forms. Two-dimensional calls are abbreviations for the three-dimensional calls, meaning the Z-coordinate of the current position is used and left unchanged when the two-dimensional form is used. If no three-dimensional calls are made, the Z-coordinate of the current position always remains 0.0.

Attributes are the characteristics of graphics figures. Color, linestyle, linewidth, and pick-ID are the general attributes of output primitives. Text size, text gap, text font, text slant, text orientation, and text justification are attributes specific to text. Each has a current value under the application program's control. The current values of applicable attributes determine the appearance of output primitives as they are generated.

When a house is drawn, its line primitives can be given a particular linestyle, such as "dashed" which is linestyle 2 on some devices, by setting the current value of linestyle to 2 before the house is created. What if the display device does not have linestyle 2? That is, what if the device cannot perform a requested attribute exactly? In these cases, AGP-3 maps the requested attribute value to a supported attribute value; for example, if linestyle 2 were not supported, the house would be drawn in linestyle 1.

In the following sections, general output primitives, general output primitive attributes, text, and text attributes are described.

General Output Primitives

Move

J2MOV	(X,Y)	Move the current position (2D, absolute)
J3MOV	(X,Y,Z)	Move the current position (3D, absolute)
JR2MV	(DX,DY)	Move the current position (2D, relative)
JR3MV	(DX,DY,DZ)	Move the current position (3D, relative)

The move routines are used to set the current position to a point in the world coordinate system. They do not produce visual output on the display device. Instead, they affect the position at which subsequent primitives will be located.

Coordinate positions can be specified as either absolute or relative with respect to the current position. Also, the two-dimensional routines are abbreviations for the three-dimensional routines where the Z-coordinate of the current position is left unchanged.

Line

J2DRW	(X,Y)	Draw a line (2D, absolute)
J3DRW	(X,Y,Z)	Draw a line (3D, absolute)
JR2DR	(DX,DY)	Draw a line (2D, relative)
JR3DR	(DX,DY,DZ)	Draw a line (3D, relative)

The line routines draw a line from the current position to a point specified in the parameters. For absolute forms, this point is the world coordinate point specified: either (X,Y) or (X,Y,Z). For relative forms, the end of the line is found by adding the given offsets to the current position. After the line is output, the endpoint is made the new current position. The following example draws the square in Figure 3-1.

```
CALL J2MOV (-0.5,-0.5)          * move to the lower left corner
CALL J2DRW ( 0.5,-0.5)          * draw a square: bottom,
CALL J2DRW ( 0.5, 0.5)          *   right side,
CALL J2DRW (-0.5, 0.5)          *   top,
CALL J2DRW (-0.5,-0.5)          *   left side
```

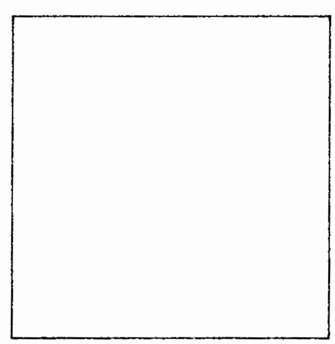

Figure 3-1. A Square Created with Moves and Draws.

The following example draws the outline of a house, as shown in Figure 3-2.

```
CALL J3MOV (-0.5,-0.5, 0.0)          * move to the lower left corner
CALL JR3DR ( 1.0, 0.0, 0.0)          * draw a house: bottom,
CALL JR3DR ( 0.0, 0.6, 0.0)          *   right wall,
CALL JR3DR (-0.5, 0.2, 0.0)          *   right side of roof,
CALL JR3DR (-0.5,-0.2, 0.0)          *   left side of roof,
CALL JR3DR ( 0.0,-0.6, 0.0)          *   left wall.
```

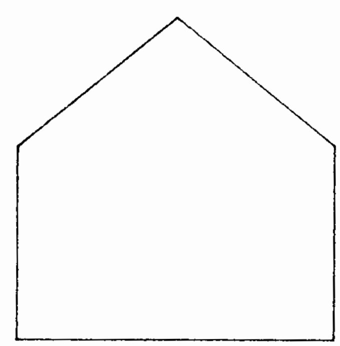

Figure 3-2. A House Created with Moves and Lines.

Polyline

J2PLY	(NPOINT,XVEC,YVEC)	Draw a polyline (2D, absolute)
J3PLY	(NPOINT,XVEC,YVEC,ZVEC)	Draw a polyline (3D, absolute)
JR2PL	(NPOINT,DXVEC,DYVEC)	Draw a polyline (2D, relative)
JR3PL	(NPOINT,DXVEC,DYVEC,DZVEC)	Draw a polyline (3D, relative)

The polyline routines are used to draw a connected line sequence. The number of points in the line sequence is found in NPOINT, while the points (or, for relative calls, the point offsets) themselves are passed as arrays in the remaining parameters. Initially, a move is made to the first point. The connected line sequence begins there, continuing to the second point, the third point ... and so on until the NPOINT point is reached.

The current position is left at the NPOINT element of the array. If NPOINT equals 1, these calls set the current position to the first element of the array and produce no lines; that is, a move is produced.

As in the move and line routines, coordinate positions can be specified as either absolute or relative to the current position. Note that the current position is updated after each vector is output, so that the relative coordinates of the next vector are calculated from this new position. Also, the two-dimensional forms are shorthand for three-dimensional forms; they leave the current Z-coordinate unchanged.

Polylines are a notational convenience for the application programmer. One polyline call may replace a number of line calls, producing a shorter but functionally equivalent application program. The following example draws the house in Figure 3-3 using only one polyline.

```
REAL X(5), Y(5), Z(5)
DATA X / 1.0, 0.0,-0.5,-0.5, 0.0/
DATA Y / 0.0, 0.6, 0.2,-0.2,-0.6/
DATA Z / 0.0, 0.0, 0.0, 0.0, 0.0/
       :
       :
CALL J3MOV (-0.5,-0.5, 0.0)         * establish current position for
                                      relative polyline call
CALL JR3PL (5,X,Y,Z)                * draw house
```

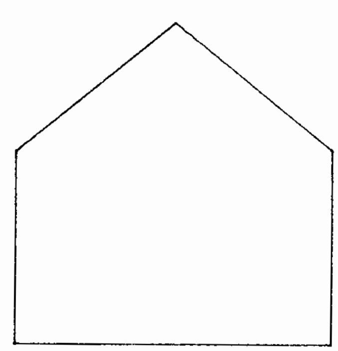

Figure 3-3. A House Created with Polyline.

Marker

J2MRK (X,Y,MARKNO)	Mark a point (2D, absolute)
J3MRK (X,Y,Z,MARKNO)	Mark a point (3D, absolute)
JR2MK (DX,DY,MARKNO)	Mark a point (2D, relative)
JR3MK (DX,DY,DZ,MARKNO)	Mark a point (3D, relative)

Markers identify individual points in the world coordinate system. Depending on a particular display device's capabilities, AGP-3 utilizes either software or hardware to generate the markers. When one of these functions is invoked, the current position is updated to the point specified by the parameters, then the desired marker symbol is output centered at that point.

A variety of markers are available, selected via the MARKNO parameter. The size and orientation of markers is fixed for a device. Unlike other output primitives, they are not affected by the viewing transformation in effect when they are generated. AGP-3 supports these 19 markers on all devices:

1 - '.'	7 - rectangle	13 - '3'
2 - '+'	8 - diamond	14 - '4'
3 - '*'	9 - rectangle with cross	15 - '5'
4 - 'O'	10 - '0'	16 - '6'
5 - 'X'	11 - '1'	17 - '7'
6 - triangle	12 - '2'	18 - '8'
		19 - '9'

In addition to these, a device may support additional markers of its own.

In the following example the bottom, center of the house is marked. The house has been created previously, with several line primitive calls or a polyline call, with the appropriate data. From now on in this chapter, the subroutine HOUSE is used to draw this series of line primitives.

```
CALL HOUSE                         * draw a house
CALL J3MRK (0.0, 0.0, 0.0, 8)      * mark it with a diamond
```

The house is shown in Figure 3-4.

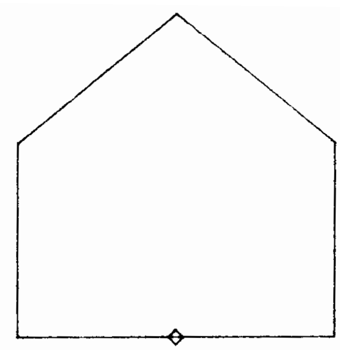

Figure 3-4. Marking the Threshold of a House.

General Output Primitive Attributes

A primitive can have several different attributes, or characteristics, all of which are defined individually. A line, for example, can have four attributes: color, linestyle, linewidth, and pick-ID, while a marker can have two: color and pick-ID. Some attributes are general, that is, they can be applied to more than one primitive, while some are specific to a single primitive. Color, for example, can be applied to lines, polylines, markers and text, while font can be applied to text only.

AGP-3 maintains a set of current attribute values for all primitive attributes. They are set to defaults at system initialization. A primitive is assigned the system-maintained attribute values when it is created. All general attribute values except pick-ID are defined as indices in device-dependent tables. An attribute value can be changed from the default index to another index with a call to the appropriate subroutine.

In the following section, the general attributes of color, linestyle, linewidth, and pick-ID are discussed. Table 3-1 shows which of these general attributes apply to the primitives line, polyline, marker, and text.

Table 3-1. General Primitive Attributes.

Attribute	LINE	POLYLINE	MARKER	TEXT
Color	X	X	X	X
Linestyle	X	X		X
Linewidth	X	X		X
Pick-ID	X	X	X	X

Color

JCOLR (COLOR) Set color index

JCOLR sets an index into a device-dependent table. With it the color of lines, polylines, markers, and text can be specified. The number of colors available is device dependent, but every device supports at least one visible color. The mapping between the value of COLOR and the available colors on a graphics device is device dependent. For example, color 1 might correspond to slot 1 on a plotter, but the pen in slot 1 could be any color. On a different device, however, color 1 might always refer to the color red.

The maximum number of colors available is 255. The default value of COLOR is 1.

Linestyle

JLSTL (LINES) Set linestyle index

JLSTL sets an index into a device-dependent table. With it the pattern used to draw lines, polylines, and text can be specified. A maximum of 255 linestyles is available, but the actual number supported by a particular graphics device is device dependent.

Linestyles use various patterns of long line segments, short line segments, and blank spaces to produce a variety of line primitive appearances. Some examples are shown in Figure 3-5.

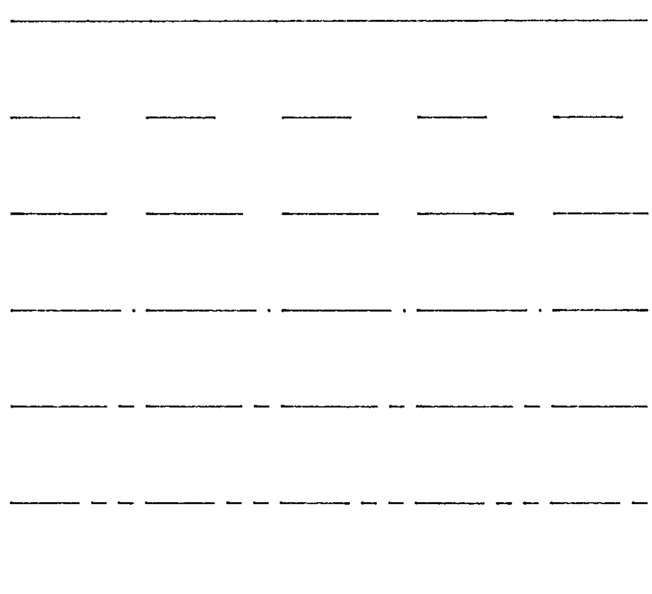

Figure 3-5. Various Linestyles.

Care should be taken when linestyles that are not entirely solid are used. For instance, if a linestyle contains a number of blank spaces, a short line primitive might produce nothing visible, since the primitive's end could be reached before the non-blank portion is encountered.

Even when longer line primitives are drawn, a linestyle may not necessarily produce the desired appearance. Some linestyles have readily identifiable line primitive endpoints, while others do not. Those that do, called vector-adjusted linestyles, are good for objects whose endpoints must be seen, such as rectangles.

Of the remaining linestyles, a distinction is made between those that begin each line primitive in the same way (start-adjusted), and those that continue a new line primitive as if the previous primitive had never ended (continuous). Continuous linestyles are good for objects whose endpoints are not to be emphasized, such as curves. (Curves are drawn by outputting many small line primitives, one after another.)

The effect of linestyle on graphics text is device dependent, as some devices allow text to be displayed in a variety of linestyles.

At AGP-3 initialization the linestyle index is 1.

Linewidth

JLWID (LWIDTH) Set linewidth index

JLWID sets an index into a device-dependent table. With it, the linewidth of lines, polylines, and text can be defined. While all devices support at least one linewidth, the maximum number of supported linewidths is device dependent. A value of 1 specifies the thinnest width possible. When multiple widths are supported, the width of the line increases as the value of LWDTH does. If LWDTH is greater than the number of line widths supported, or if linewidth is less than 1, the linewidth is set to the thinnest available width.

Text is affected by linewidth only if the device is capable of applying it to text. Markers are not affected by this call as they are defined to be output with the thinnest linewidth supported by the device.

At initialization, the default linewidth is 1.

Pick-ID

JPKID (PICKID) Set pick identifier

JPKID sets the pick-ID attribute. It is used to give different names to different parts of a segment. In this way, a segment can be defined with several identifiable components.

When a primitive is created, the current pick-ID is bound to it. Its pick-ID and the name of the segment to which it belongs are returned to the application program whenever the primitive is selected by the operator using the pick input device. Pick-ID applies only to primitives within a segment. Primitives outside a segment do not have a pick-ID associated with them and are not detectable by the pick input function. The initial pick-ID attribute is 1.

Figure 3-6 represents a menu of candies. If the menu were identified as a segment, and each type of candy were given a separate pick-ID, each type of candy could be selected separately from the others.

Jelly Beans

Gum Drops

Candy Canes

Figure 3-6. A Menu of Candies.

Text

AGP-3 supports three different levels of graphics text. High quality text strictly adheres to all user requested text attributes and therefore is slowest to produce. Low quality text ignores or loosely applies certain attributes and therefore is fastest to produce. Medium quality text is between the other two both in terms of accuracy of representation and the time required to produce it.

The concept of a character cell is needed to discuss text. As shown in Figure 3-7, it is a box that defines the size of an individual text character. It generally contains the entire character, but in some cases, for example "y", part of the character may extend outside the character cell.
Text fonts are described as either fixed or variable-width, depending on whether the character cells of different characters have the same or different widths.

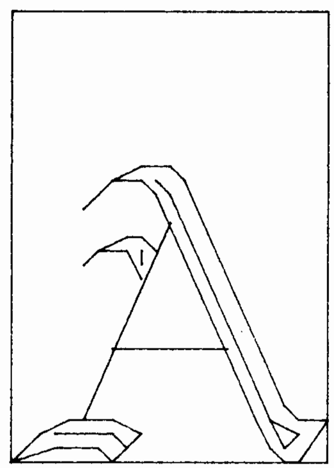

Figure 3-7. Character Cell.

High Quality

JTEXH (NCHARS,STRING) Output high quality text

Each high quality character is formed using a series of lines, called strokes. The coordinates for these strokes are supplied in a font file. For high quality text, all text attribute specifications are strictly applied. The strokes that make up the characters are subject to the viewing transformation in the same way as line primitives.

After outputting a high quality text string containing NCHAR characters, the current position is updated to the lower left corner of where the next character cell would have been, had there been another character output. Figure 3-8 shows the current position before and after a string is output.

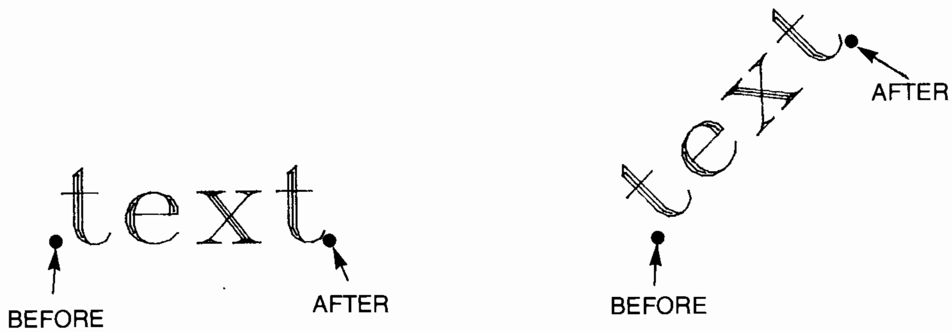

Figure 3-8. Text and Updating the Current Position.

Medium Quality

JTEXM (NCHARS,STRING) Output medium quality text

Medium quality characters are output using the hardware character generator in the graphics display device. If no hardware character capabilities exist, software at the device handler level will simulate hardware-generated text. The orientation of medium quality text is always upright. AGP-3 positions each character individually and uses the hardware character generator (or simulator) to output the characters one at a time. Once the entire string has been output, the current position is updated to the same position at which high quality fixed-width text would have left it (see Figure 3-8).

Medium quality text is affected by the viewing transformation as follows. The lower left corner of each character cell is placed in the same spot as if high quality text was being output, but the characters are all vertical and their sizes chosen on a "smaller-best-fit" basis.
Smaller best fit is defined to be:

1. the largest character whose height is less than or equal to the requested cell height and whose width is less than or equal to the requested cell width, or,

2. if even the minimum hardware size does not meet criterion 1, the minimum character size is used.

Low Quality

JTEXL (NCHARS,STRING) Output low quality text

Low quality characters are output using the hardware character generator if one exists. As with medium quality text, device handler software will simulate hardware character generation if one does not exist. The size of low quality text is also determined using the "smaller-best-fit" algorithm. AGP-3 then generates the entire string as a unit. Therefore, the start of the string is correctly placed but individual characters may not be. After outputting a low quality text string, the current position is updated to the same position at which fixed-width high quality text would have left it (see Figure 3-8).

Text-Specific Attributes

The following section describes attributes specific to the text primitive. As shown in Table 3-3, the number of text attributes available depends on the level of text used.

Table 3-3. Text-specific Attributes

Text Attribute	LOW TEXT	MEDIUM TEXT	HIGH TEXT
Size	X	X	X
Gap		X	X
Orientation		X	X
Justification	X	X	X
Font			X
Slant			X

Size and Gap

JCSIZ (WIDTH,HEIGHT,GAP) Define character size and gap

JCSIZ sets the size attribute for subsequently created text. WIDTH and HEIGHT specify the world coordinate size of a character cell. GAP specifies the spacing between character cells.

For low and medium quality text the actual character size is determined on a "smaller-best-fit" basis. Low quality text size is determined only once for the entire string, while medium quality text size is recomputed for each character so that the viewing transformation is more exactly applied. For high quality text with a fixed-width font, each character maps exactly to the requested size. For high quality text with a variable-width font, the letter "A" maps to the requested size, and the width of the other characters varies according to their widths in the font. This is shown in Figure 3-9.

JCSIZ (0.035, 0.05, 0.0)

JCSIZ (0.07, 0.05, 0.0)

JCSIZ (0.07, 0.1, 0.0)

JCSIZ (0.035, 0.1, 0.0)

Figure 3-9. Various Text Widths and Heights.

GAP, which determines the spacing between character cells, is expressed as a fraction of character cell width. A gap of 0.0 produces "normal" spacing. A positive value spreads characters out, while a negative value brings the characters closer together. Figure 3-10 shows the effect of gap on high quality text.

JCSIZ (0.07, 0.1, 0.0)

JCSIZ (0.07, 0.1, 0.4)

JCSIZ (0.07, 0.1, 1.0)

JCSIZ (0.07, 0.1, -0.2)

Figure 3-10. Various Text Gaps.

Gap has no affect on character spacing for low quality text, but it does affect the placement of the current position at the completion of the text call. To understand this, remember that low quality text is sent to the device a string at a time; the device controls the placement of individual characters. After the string is sent, the current position is updated just as if fixed-width high quality text had been used. That is, the current position is updated exactly according to the requested width and gap independent of the actual results on the device, meaning gap will affect where the next string begins, but has no affect on this string.

Contrast this with medium quality text, which is sent to the device on a character-by-character basis. After the first character is sent, the gap is applied in determining the placement of the current position. Then the next character is placed starting at the new current position, the current position is updated, the third character is placed, and so on. The result here is that gap affects both the position of characters in this string and where the next string will begin.

High quality text, of course, is sent to the device neither on a string nor character basis, but "stroke-by-stroke". Thus width and gap exactly determine both the position of characters in this string and the beginning position of the next string.

The initial values are 0.05 for the height and 0.035 for the width in world coordinate units. Gap is initially 0.0, meaning additional space will neither be added nor subtracted from between character cells.

Orientation

JCORI(XBAS,YBAS,ZBAS,XPLN,YPLN,ZPLN)   Define character orientation

JCORI specifies the character base vector and the character plane vector in world coordinates. The character base vector is (XBAS,YBAS,ZBAS) and the character plane vector is (XPLN,YPLN,ZPLN). Both vectors are relative to the origin.

The character base vector determines the direction of textual output; that is, the bottoms of characters are lined up along this vector. The base of each character cell extends in the same direction as the character base vector; the sides of each character cell lie in the same plane as the character base and character plane vectors.

Another vector is derived from these two character vectors. It is called the character up vector, and is simply that vector perpendicular to the character base vector lying in the same plane as the character plane vector. This vector describes the up direction of characters (see Figure 3-11).

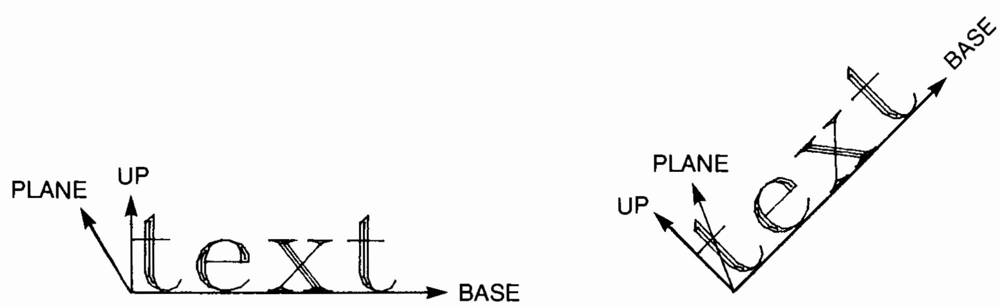

Figure 3-11. Character Vectors: Base, Plane, and Up.

The effects of these vectors are summarized in Figure 3-12. The plane vector is only relevant to high quality text - the other two ignore it. The base vector, as would be expected, is relevant to high quality text, but it also has a limited effect on medium quality text, as follows: the current position is updated character-by-character along it, but the characters themselves do not have their bases parallel to it. No character vectors affect low quality text.

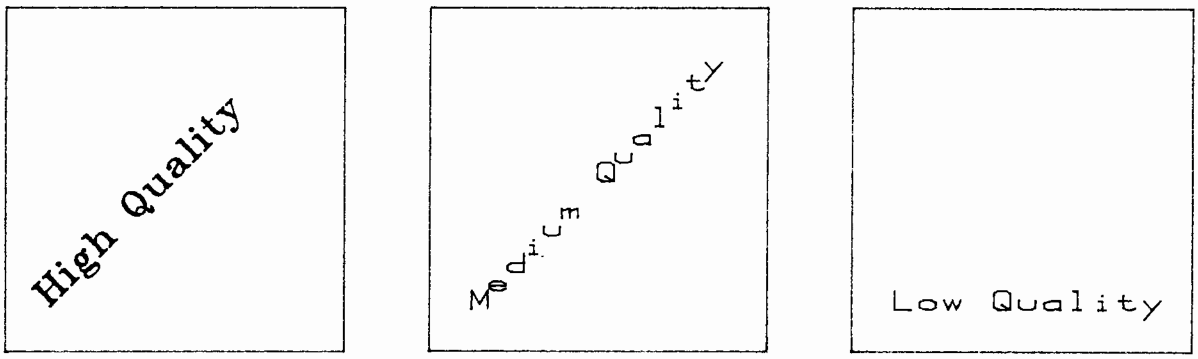

Figure 3-12. Effect of Character Orientation.

The initial character base is (1.0,0.0,0.0) in world coordinates. The initial character plane is (0.0,1.0,0.0). Hence, characters initially appear parallel to the X-Y plane with their bases advancing along the positive X-axis.

Justification

JJUST (BASE,UP) Set the text justification

JJUST determines the placement of the entire text string relative to the current position. A text string may be justified both horizontally (along the character base vector) and vertically (along the character up vector). Conceptually, justification may be thought of as putting an imaginary box around the entire text string and then moving the box and the text string the required amount.

BASE is expressed as a fraction of the total string width, and UP is expressed as a fraction of the total string height. Their initial values are both 0.0. BASE = 0.0 implies that 0% of the width of the text string is to be placed on the side of the current position opposite the character base vector. UP = 0.0 implies that 0% of the height of the character cell is to be placed on the side of the current position opposite the character up vector. If the character up vector point "up" and the character base vector points "right", then the initial value of (0.0, 0.0) implies that text strings are output with the current position at the lower left corner of the text string.

The example below produces the text in Figure 3-13.

```
CALL JJUST (0.0, 0.0)            * use default justification
CALL J2MRK (0.0, 0.6, 3)         * set and mark current position
CALL JTEXH (12, 12HDefault Just)  * output high quality text
      :
CALL JJUST (0.5, 0.0)            * horizontally justified
CALL J2MRK (0.0, 0.3, 3)         *
CALL JTEXH (12, 12HHorizontally)  *
      :
CALL JJUST (0.0, 0.5)            * vertically justified
CALL J2MRK (0.0, 0.0, 3)         *
CALL JTEXH (10, 10HVertically)    *
      :
CALL JJUST (0.5, 0.5)            * justified both ways
CALL J2MRK (0.0,-0.3, 3)         *
CALL JTEXH (8, 8HCentered)        *
      :
CALL JJUST (1.0, 1.0)            * upper, right corner
CALL J2MRK (0.0,-0.6, 3)         *
CALL JTEXH (11, 11HUpper Right)   *
```

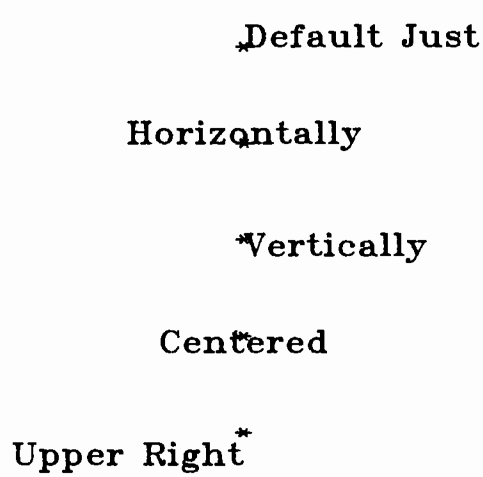

Figure 3-13. Text Justification.

Font and Slant

```
JDFNT(FONT,SLANT,FNAME,SEC,CRN,CONTRL)     Define a font
JFONT(FONT)                                Set the current font
```

The JDFNT call associates a font number with a high quality text font file. FONT is the number by which that font will be subsequently identified. The SLANT parameter specifies the angle to slant each character as it is drawn. FNAME, SEC and CRN specify the name, security code, and cartridge of the font file (a 0 in CRN means search all cartridges for FNAME). The CONTRL word is used to control characteristics of character production or font file usage.

JFONT is used to set the current font number for high quality text calls. Initially the current font = 0, signifying that there is no current font. A value other than 0 will set that font to be the current font for subsequent high quality text calls. At initialization no font files are defined.

The following example produces the text fonts in Figure 3-14.

```
CALL JDFNT (1, 0.0, 6HFONT1 , 0, 0, 0)    * open font file 1
CALL JDFNT (2, 0.0, 6HFONT2 , 0, 0, 0)    * open font file 2
CALL JDFNT (3, 0.0, 6HFONT3 , 0, 0, 0)    * open font file 3
CALL JDFNT (4, 0.0, 6HFONT4 , 0, 0, 0)    * open font file 4
CALL JDFNT (5, 0.0, 6HFONT5 , 0, 0, 0)    * open font file 5
CALL JDFNT (6, 0.0, 6HFONT6 , 0, 0, 0)    * open font file 6
       :
       :
CALL J2MOV (-0.9, 0.6)                   * set current position
CALL JFONT (1)                           * current font is font 1
CALL JTEXH (9, 9HEurostyle)              * draw high quality text
       :
CALL J2MOV (-0.9, 0.3)                   * move current position
CALL JFONT (2)                           * current font is font 2
CALL JTEXH (13, 13HSimplex Roman)        *
       :
CALL J2MOV (-0.9, 0.0)                   *
CALL JFONT (3)                           * font 3
CALL JTEXH (13, 13HTriplex Roman)        *
       :
CALL J2MOV (-0.9,-0.3)                   *
CALL JFONT (4)                           * font 4
CALL JTEXH (6, 6HScript)                 *
       :
CALL J2MOV (-0.9,-0.6)                   *
CALL JFONT (5)                           * font 5
CALL JTEXH (12, 12HMathematical)         *
       :
CALL J2MOV (-0.9,-0.9)                   *
CALL JFONT (6)                           * font 6
CALL JTEXH (6, 6HGothic)                 *
```

Eurostyle

Simplex Roman

Triplex Roman

Script

ματηεματιψαω

Gothic

Figure 3-14. Text Fonts.

The SLANT parameter specifies the angle in radians to slant each character as it is drawn. The following example produces the various slants in Figure 3-15.

```
CALL JDFNT (1, 0.0, 6HFONT1 ,0,0,0)      * open font file, no slant
CALL JDFNT (2, 0.2, 6HFONT1 ,0,0,0)      * same file, slight slant
CALL JDFNT (3, 1.0, 6HFONT1 ,0,0,0)      * same file, more slant
CALL JDFNT (4,-1.0, 6HFONT1 ,0,0,0)      * same file, slant backwards
CALL JDFNT (5,-0.2, 6HFONT1 ,0,0,0)      * same file, less backslant

CALL JCSIZ (0.07, 0.1, 0.2)              * set character size and gap
          :
          :
CALL JFONT (1)                           * no slant
CALL J2MOV (-0.5, 0.6)                   * set current position
CALL JTEXH (13, 13HDefault Slant)        * draw high quality text

CALL JFONT (2)                           * slight forward slant
CALL J2MOV (-0.5, 0.3)                   * set current position
CALL JTEXH (10, 10HItalicized)           * draw high quality text

CALL JFONT (3)                           * more forward slant
CALL J2MOV (-0.5, 0.0)                   * set current position
CALL JTEXH (12, 12HFull Italics)         * draw high quality text

CALL JFONT (4)                           * opposite direction
CALL J2MOV (-0.5,-0.3)                   * set current position
CALL JTEXH (14, 14HBackward Slant)       * draw high quality text

CALL JFONT (5)                           * still opposite direction
CALL J2MOV (-0.5,-0.6)                   * set current position
CALL JTEXH (14, 14HLess Backslant)       * draw high quality text
```

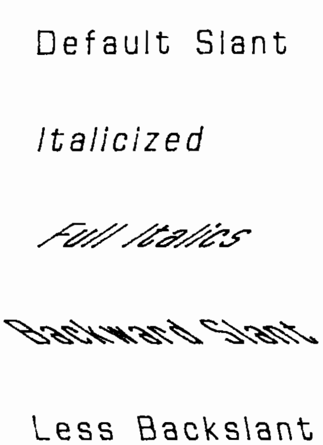

Figure 3-15. Text Slants.

Alphanumeric Display

JALPH (ID,STRING,NCHARS)   Output to the alphanumeric display

JALPH allows the application program to send non-graphics text and data to an alphanumeric display device. This call can be used to send prompts, give status or print other messages.

ID identifies the work station associated with the alphanumeric display device. The text in STRING should be NCHARS long. A carriage return line feed (CRLF) is attached to the end of STRING when it is sent to the device, unless the last character in STRING is an underscore. This suppresses the CRLF. If the alphanumeric device is physically the same device as the graphics display device, any escape codes which affect the graphics application should not be passed in STRING, since they may place the graphics display device in an unknown state.

JALPH's output is not included in segments. This is because JALPH does not create graphics output primitives, and only graphics output primitives are saved in segments.

Chapter 4
Viewing Transformations

General

A viewing transformation is the method by which two or three-dimensional objects are turned into viewable images. Viewing transformations modify objects for viewing by reducing them from three dimensions to two dimensions and converting them to device-dependent units.

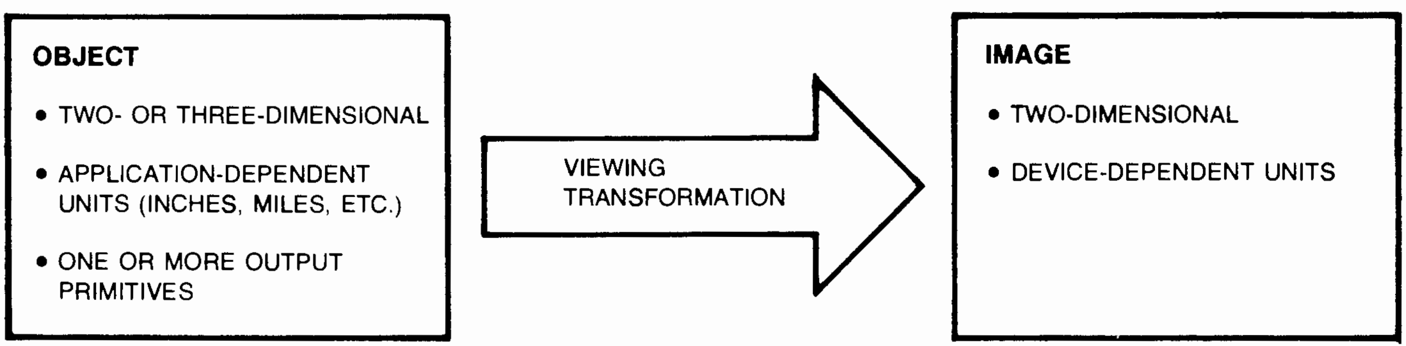

Figure 4-1. The Viewing Transformation.

AGP-3 treats two-dimensional viewing and three-dimensional viewing in an integrated fashion; the two-dimensional case is a subset of the three-dimensional case. The viewing capabilities are such that two-dimensional users need not know about three-dimensional constructs. Since the two-dimensional case is much easier to explain and use than the three-dimensional one, the following description is broken into two parts.

Two-Dimensional Viewing Transformation

A number of concepts are needed to understand how AGP-3 turns two-dimensional objects in the world coordinate system into images on a display device. The steps in this transformation are summarized in Figure 4-2.

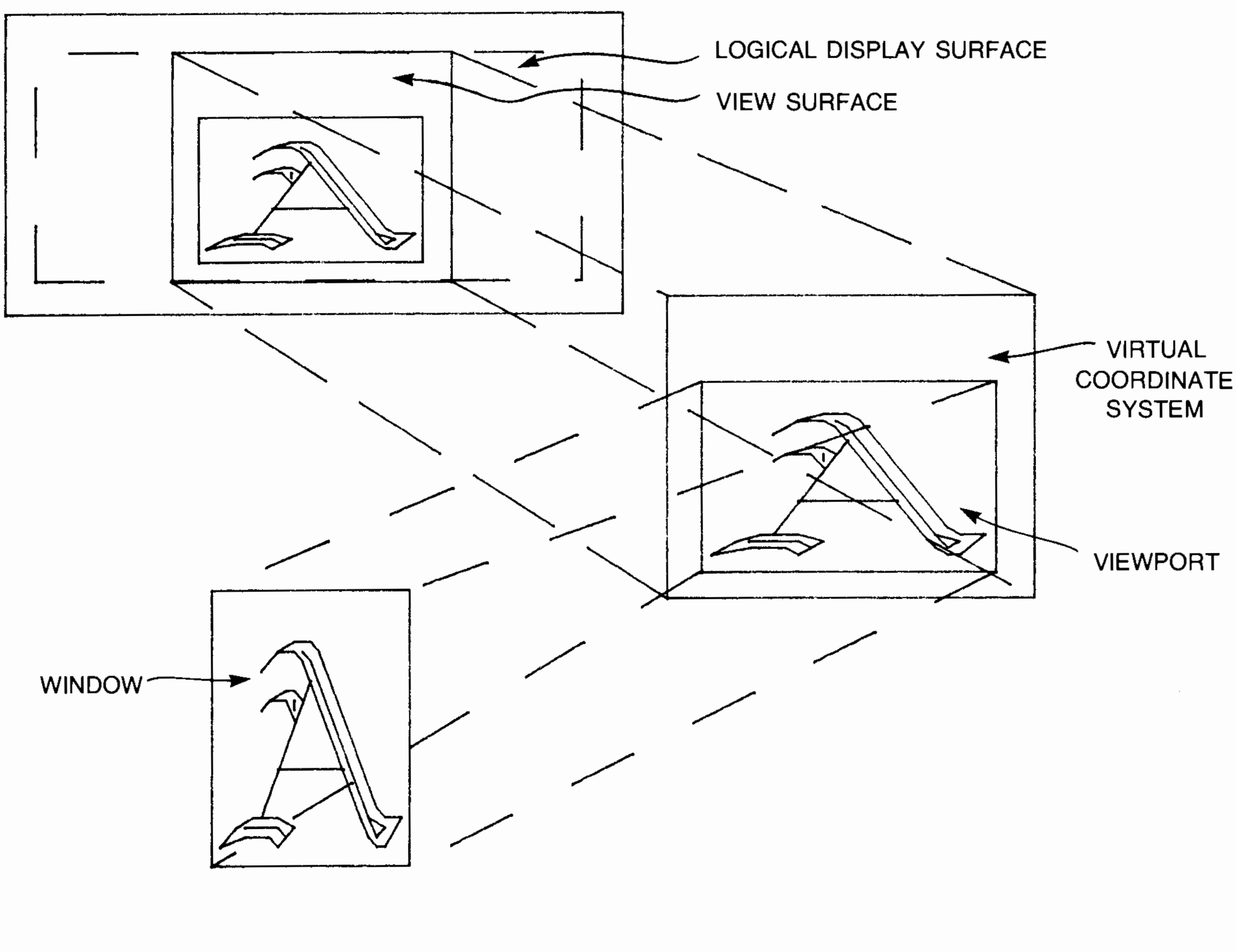

Figure 4-2. The Two-Dimensional Viewing Transformation.

Logical display limits allow a subset of a display device, known as the logical display surface, to be used for output. On a plotter, the bed of the plotter may be 500 millimeters by 1000 millimeters, while the paper in use might be only 250 millimeters square. The logical display limits of the plotter could be set to that size, so that AGP-3 would treat the plotter from that point on as if it were physically 250 millimeters square.

The logical display limits, in turn, determine the size of the <u>view surface</u>. The view surface's dimensions are determined by:

1. the aspect ratio of the virtual coordinate system, and

2. the logical display limits of the display device.

Recall that the virtual coordinate system is the two-dimensional system whose units range from 0.0 to 1.0. World coordinate data is translated to virtual coordinates, allowing the user program to remain device independent. However, not all devices on which graphics is produced have the same aspect ratio (height-to-width ratio). If the virtual coordinate system were always a unit square, a non-square logical display surface (one that is rectangular) would not be entirely available. To make all of a non-square logical display surface available, AGP-3 allows the aspect ratio of the virtual coordinate system to be changed as needed.

The view surface is the largest rectangle within the logical display limits having the aspect ratio of the virtual coordinate system. Once the view surface is determined, a portion of it is designated the viewport. An application might use several viewports, perhaps to show different views of the same object.

The window defines what objects will become images. In Figure 4-2, the window just encompasses the letter "A". Objects whose primitives lie outside the window can be <u>clipped</u> (excluded from view) and not mapped to the viewport. When the aspect ratio of the window does not match that of the viewport, this mapping may cause distortion; that is, the images of the primitives will appear too "fat" or "thin". Distortion is deliberately used in Figure 4-2 to differentiate between the window and viewport.

Display Limits

JDLIM (ID,XMIN,XMAX,YMIN,YMAX)   Set logical display limits

JDLIM specifies the area of the display device on which output may be produced by setting limits. Logical display limits are useful for excluding a portion of the display device from receiving images. XMIN, XMAX, YMIN, and YMAX are described in millimeters as offsets from the device's origin (the origin of a device is usually in its lower-left corner). The work station containing the device is specified via the ID parameter, which must be an integer between 1 and 8. (Refer to Chapter 7 and the AGP-3 Reference Manual for more information about work stations.)

JDLIM implicitly redefines the view surface to be the maximum rectangle within the logical display limits that has the same aspect ratio as the virtual coordinate system.

Aspect Ratio

JASPK (XSIZE,YSIZE) Set aspect ratio of virtual coordinate system

JASPK sets the aspect ratio of the virtual coordinate system (and hence the aspect ratio of the view surface) to be YSIZE divided by XSIZE. A ratio of 1.0 specifies a square virtual coordinate system, a ratio greater than 1.0 specifies it to be higher than it is wide, and a ratio less than 1.0 specifies it to be wider than it is high. Since XSIZE and YSIZE are used to form a ratio, they may be in any convenient units (so long as both are specified in the same units).

Based on the aspect ratio, AGP-3 calculates the range of the X and Y coordinates in the virtual coordinate system. The coordinates of the longer side always range from 0.0 to 1.0, and the coordinates of the shorter side range from 0.0 to a value that achieves the desired aspect ratio. JASPK is said to define the <u>limits</u> of the virtual coordinate system, as shown in Table 4-1.

Table 4-1. Virtual Coordinate System Limits.

Aspect Ratio (AR)	X Limits	Y Limits
AR < 1.0	(0.0, 1.0)	(0.0, AR)
AR = 1.0	(0.0, 1.0)	(0.0, 1.0)
AR > 1.0	(0.0, 1.0/AR)	(0.0, 1.0)

When JASPK is called, the viewport is automatically set to map to the entire view surface. For example, the HP 2648A Graphics Terminal has an aspect ratio of 359 / 719 = 0.4993. To utilize the entire visible screen, the application program may:

```
REAL XSIZE, YSIZE          *
      :                    * width of HP2648A is 719 device units
XSIZE = 719.0              *   and height is 359 device units
YSIZE = 359.0              * set aspect ratio of virtual
CALL JASPK (XSIZE,YSIZE)   *   coordinate system to match
      :
```

The initial aspect ratio of the virtual coordinate system is 1.0, meaning the virtual coordinate system is a unit square. This produces a view surface that is the largest inscribed square within the logical display limits. By changing the aspect ratio, the view surface can become the largest inscribed rectangle within the logical display limits.

Viewport

JVIEW (VPXMIN,VPXMAX,VPYMIN,VPYMAX) Set viewport

JVIEW sets the viewport. VPXMIN, VPXMAX, VPYMIN, and VPYMAX are specified in virtual coordinates and must be within the limits of the virtual coordinate system defined by JASPK. The initial viewport is:

(VPXMIN = 0.0; VPXMAX = 1.0, VPYMIN = 0.0; VPYMAX = 1.0)

The following program section uses two viewports. The HOUSE subroutine has been described previously; it draws a house using line primitives. The output is shown below.

```
  CALL JVIEW (0.0, 0.5, 0.0, 0.5)     * set viewport to left half
  CALL HOUSE                          *   and draw a house
        :
  CALL JVIEW (0.5, 1.0, 0.0, 0.5)     * set viewport to right half
  CALL HOUSE                          *   and draw house again
        :
```

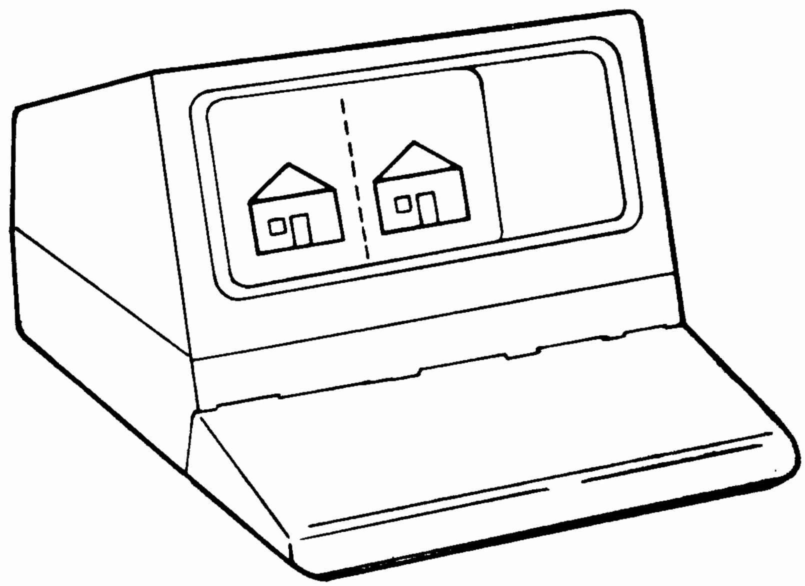

Figure 4-3. Two Viewports.

Window

JWIND (WXMIN,WXMAX,WYMIN,WYMAX) — Set window
JCLPW (OPCODE) — Control window clipping

The window is set via JWIND. For two-dimensional applications, the window is set in the X-Y plane of the world coordinate system. This is not true of three-dimensional applications, whose window is set in the viewplane. (For two-dimensional applications, these are the same, since the window is set by default in the X-Y world coordinate plane, which is the viewplane.)

The following example shows the window and viewport being set to avoid distortion.

```
CALL JVIEW (0.0, 1.0, 0.0, 0.2)        * viewport is rectangle
CALL JWIND (100.0, 200.0, 0.0, 20.0)   * window is a rectangle
```

The window's aspect ratio should be the same as the aspect ratio of the viewport to avoid distortion. In general, setting the window by calculating its dimensions as a function of the viewport is a good way to prevent distortion. In this way, no assumption is made about a previously set viewport, helping to ensure that the visual results will be those desired.

JCLPW enables (OPCODE = 1) or disables (OPCODE = 0) the window clipping function. Clipping is desirable when primitives may extend beyond the window. If clipping is suppressed and primitives <u>do</u> exceed window boundaries, the resulting image is unpredicatable.

The window is initially:

(WXMIN = -1.0; WXMAX = 1.0; WYMIN = -1.0; WYMAX = 1.0)

This produces a window with an aspect ratio of 1.0, which is the same as the aspect ratio of the initial viewport. Window clipping is initially enabled. It is strongly recommended that window clipping be left enabled, at least until a program has been fully debugged and all primitives are known to lie completely within the window. At this point, some small performance gain may be realized by disabling window clipping.

Three-Dimensional Viewing Transformation

Although three-dimensional viewing transformations are conceptually a difficult part of AGP-3, they are very powerful in nature. With relatively little application code, displayed images can be given a variety of perspectives and appearances.

AGP-3 uses a viewplane, a plane on which a three-dimensional object is projected to a two-dimensional image by passing lines called projectors, one through each point of the object, to a single point called the center of projection. When the center of projection is a finite distance from the viewplane, a perspective projection is obtained. When it is at infinity, that is, when the projectors are all parallel, a parallel projection is obtained.

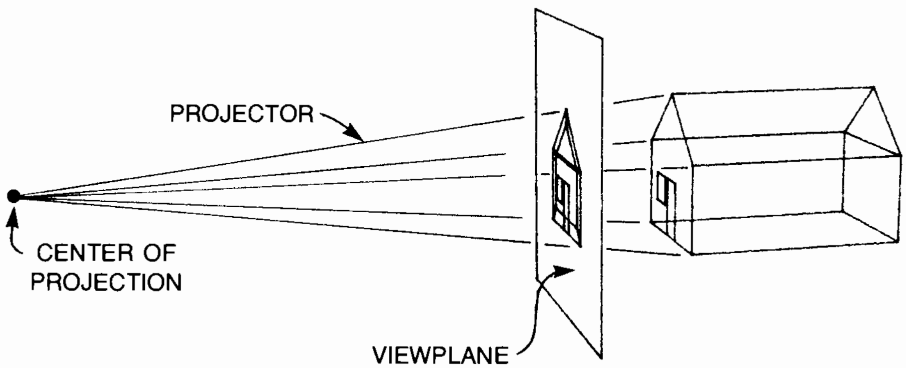

Figure 4-4. Components of a Projection.

Once a three-dimensional object is projected onto the viewplane, it then has only two dimensions, so the remaining steps are those of the two-dimensional viewing transformation: the projected primitives are clipped to a window on the viewplane, the window is mapped to the viewport, and the viewport is displayed on the display device.

A step has been omitted, however. In addition to window clipping in the viewplane, AGP-3 has depth clipping. Depth clipping clips objects to the region between two planes, the hither plane and the yon plane, which are parallel to the viewplane. Primitives lying in front of the hither plane and beyond the yon plane are eliminated from view and not projected onto the viewplane.

Together with the window, these planes define a clipping volume. This is a region of the world coordinate system outside of which primitives will not be seen on the display device. In the figure, the clipping volume includes the entire house.

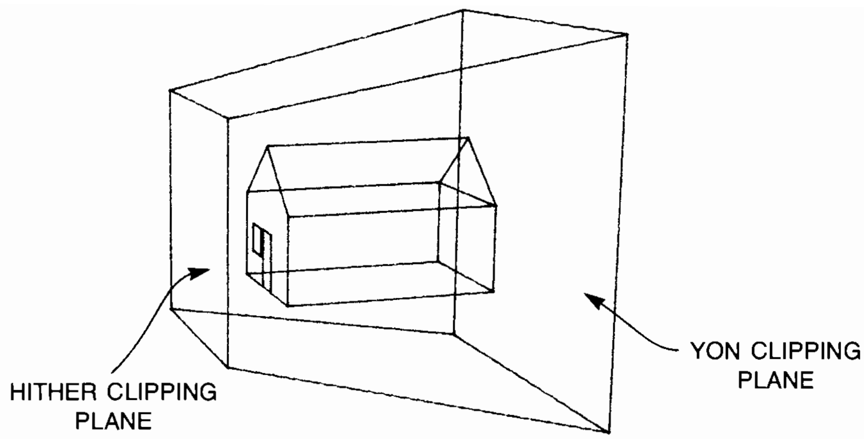

Figure 4-5. The Clipping Volume.

Viewplane

JVREF (X,Y,Z)	Define the view reference point
JVDIS (DIST)	Set viewplane distance
JVPLN (XNRM,YNRM,ZNRM,XUP,YUP,ZUP)	Orient the viewplane

The first step in the three-dimensional viewing transformation is to establish a view reference point with a call to JVREF. The view reference point is a world coordinate point on or near the object being viewed. It can be the center of interest, or near the center of interest of the picture. The viewing transformation is set in relation to the view reference point, facilitating later viewing modifications by simply changing its location.

Once the view reference point is set, the viewplane must be positioned. This requires two steps. First, the viewplane distance is specified with JVDIS: how far is the viewplane from the view reference point? Next, the viewplane normal is set with JVPLN: what vector is the viewplane perpendicular to? The viewplane normal extends from the viewplane through the view reference point parallel to the direction vector (XNRM,YNRM,ZNRM). Together, the viewplane distance and viewplane normal completely determine the viewplane's position.

At this point a third coordinate system, called the "viewing coordinate system", must be introduced. The viewing coordinate system is a three-dimensional system comprised of a U-axis, a V-axis, and an N-axis. These axes are established and oriented in a step-by-step fashion. The first step, positioning the N-axis, has already been taken. The N-axis is simply the viewplane normal itself. The origin of the viewing coordinate system is the point where the viewplane normal pierces the viewplane. To orient the two remaining axes requires one more vector: the view up vector.

Which way is up? This is what the view up vector tells AGP-3. Think of the window which will be used later: the view up vector keeps the window "upright" in the viewplane, affecting the image on a display device as shown.

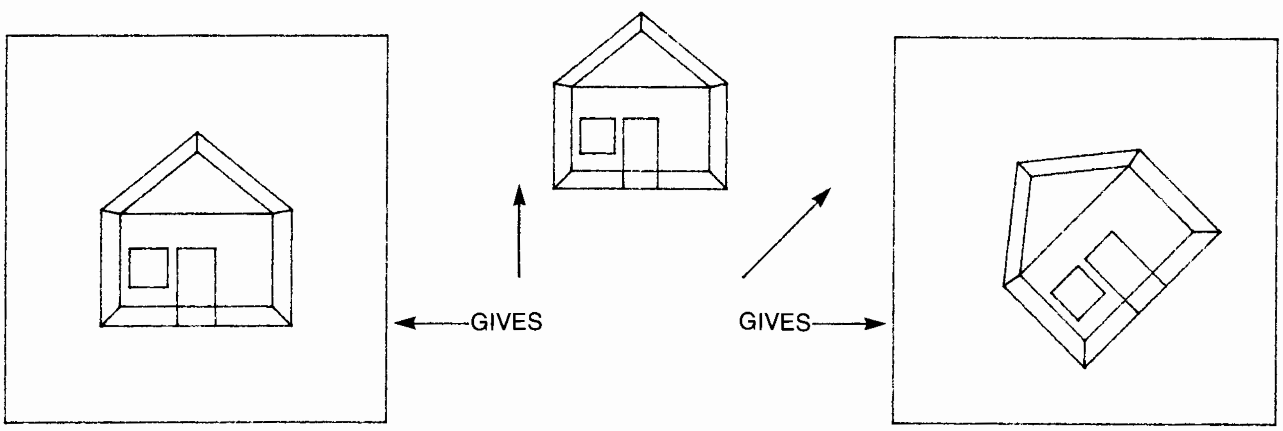

Figure 4-6. Effect of the View Up Vector.

The view up vector extends from the view reference point parallel to the direction vector (XUP,YUP,ZUP). AGP-3 projects it onto the viewplane to get the V-axis of the viewing coordinate system. (The view up vector and viewplane normal may not be parallel, or the projection will not produce a V-axis.)

Once the N and V-axes have been determined, the U-axis falls out quite nicely: it is the only possible axis that is perpendicular to the other two so as to form a left-handed coordinate system. The viewing coordinate system is thus fully defined. The positive N-axis is the viewplane normal, the positive V-axis is the projected view up vector, and the U-axis is perpendicular to both the N and V-axes so as to form a left-handed coordinate system. The origin is the point where the viewplane normal pierces the viewplane.

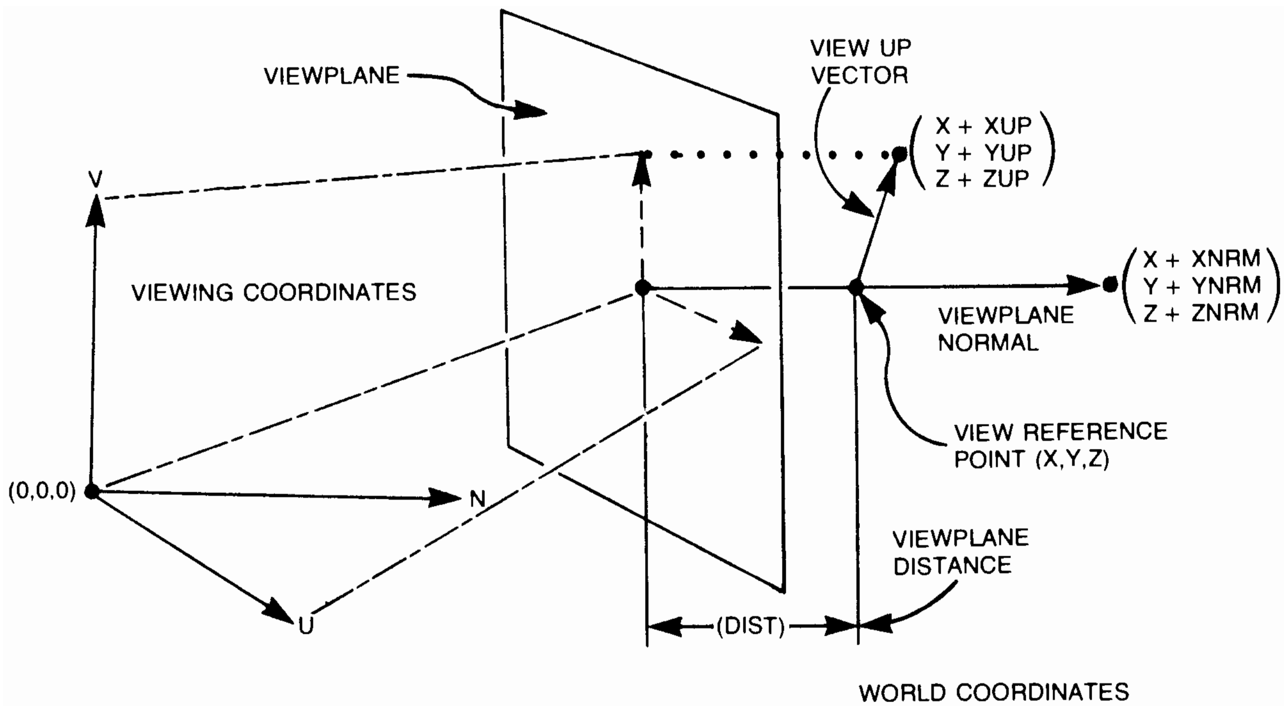

Figure 4-7. Establishing the Viewing Coordinate System.

The following example shows how the viewplane distance and viewplane normal may be manipulated. The subroutine HOUSE creates a three-dimensional house. A previously created house was two-dimensional; it is used as the front and back of this new version. Additional line primitives form the sides, window and door to complete the version shown here.

```
CALL JVREF (0.0,0.0,0.0)            * world coordinate origin
                                    *  is view reference point
CALL JVDIS (1.0)                    * viewplane is 1 unit away
CALL JVPLN (0.0,-0.3,1.0,0.0,1.0,0.) * normal gives a good angle
                                    *  and view up is Y-axis
CALL HOUSE                          * draw a 3D house
       .
       .
```

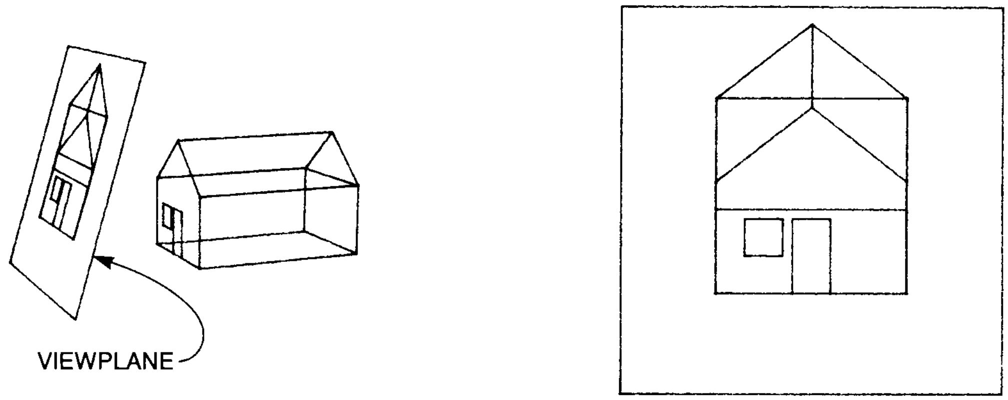

Figure 4-8. A Three-Dimensional House.

Initially, the viewplane normal is (0.0,0.0,1.0) and the view up vector is (0.0,1.0,0.0). Since the default view reference point is the world coordinate origin and the default view distance is 0.0, the viewing coordinate system is intitially coincident with the world coordinate system. This lets two-dimensional applications make the simplifying assumption that the viewing coordinate system is the world coordinate system.

Projection

JPROJ (OPCODE,DU,DV,DN)   Specify projection type and center

The type of projection may be either parallel (OPCODE = 0) or perspective (OPCODE = 1). Recall that parallel projections are really just a special case of perspective projections, where the center of projection is infinitely far from the viewplane. OPCODE distinguishes between these two cases to allow use of a specialized, more efficient algorithm for parallel projections.

```
CALL JVDIS (1.0)               * set viewplane distance of 1 unit
CALL JPROJ (1,0.0,0.0,-4.0)    * select perspective projection to
                               *  a point 4 units from the viewplane
CALL HOUSE                     * draw a 3D house
```

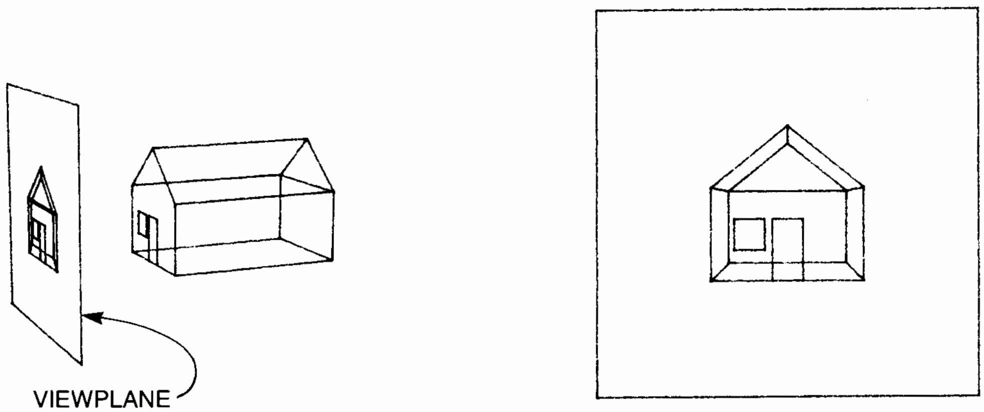

Figure 4-9. Perspective Projection of House.

The viewing coordinate system is used to specify the next feature of the viewing transformation: the center of projection or parallel projector. When the projection type is perspective, (DU,DV,DW) are the coordinates of a point in the viewing coordinate system which defines the center of projection. This is the point to which projectors from each point on the object converge.

When the projection type is parallel, the projectors are actually parallel to a line between (DU,DV,DN) and the origin of the viewing coordinate system.

When DU and DV are both zero, the projectors are perpendicular to the viewplane, forming a true parallel projection. When one or both of DU and DV are non-zero, an oblique parallel projection is defined, meaning the projectors intersect the viewplane at an oblique angle (see Figure 4-10).

Where can the center of projection or parallel projector be located? It must be in front of the viewplane - that is, on the opposite side of the viewplane from the object to be viewed (and therefore the view reference point). Otherwise, the projectors would not pass through the viewplane, and no image would be formed. This translates to the rule, DN must always be negative.

```
CALL JVDIS (1.0)               * set viewplane distance of 1 unit
CALL JPROJ (0,0.3,0.3,-1.0)    * oblique parallel projection with
                               *   projectors at 45 degree angle
                               *   to all axes
CALL HOUSE                     * draw a 3D house
```

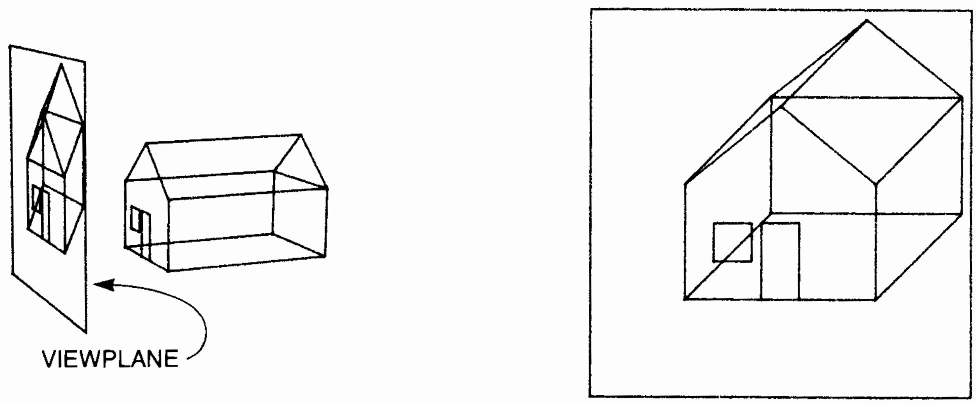

Figure 4-10. Oblique Parallel Projection of House.

The initial projection is parallel, with center of projection (0.0,0.0,-1.0). Interestingly, this makes three-dimensional objects that are symmetric in the Z direction appear two-dimensional, because the primitives farther away from the viewplane are hidden by those nearer it. Changing the projection type to perspective will allow all primitives to be seen.

Depth Clipping

JCLPD (HITHR,YON)	Control hither and yon clipping
JDPTH (HDIST,YDIST)	Position hither and yon clipping planes

JCLPD allows the application program to enable or disable hither and yon clipping. JDPTH determines where that clipping will occur. The depth clipping planes are always parallel to the viewplane; HDIST and YDIST specify the distances of the hither and yon planes, respectively, from the viewplane in viewing coordinates. The hither and yon planes are always on the opposite side of the viewplane from the center of projection, and the yon plane is always farther from the viewplane than the hither plane.

```
CALL JVREF (0.0, 0.0, 1.0)       * place view reference point
CALL JVDIS (2.0)                 *  and set viewplane distance
CALL JPROJ (1, 0.0, 0.0,-4.0)    * projection is perspective
CALL JCLPD (1, 1)                * turn on hither and yon clipping
CALL JDPTH (0.0, 2.0)            * set distances from viewplane
CALL HOUSE                       * draw house, back third clipped
```

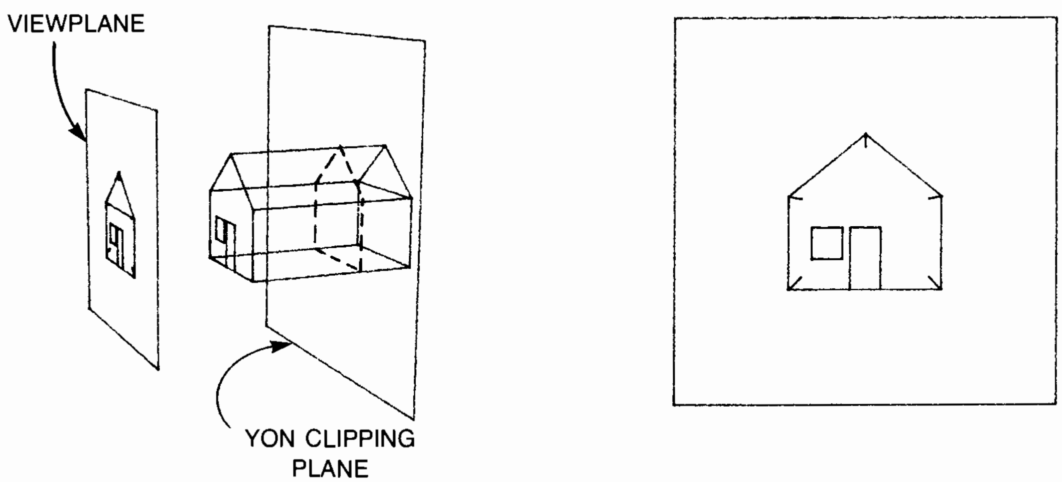

Figure 4-11. House With Yon Clipping.

At AGP-3 initialization, hither clipping is enabled while yon is disabled. Both HDIST and YDIST are 0.0, meaning hither clipping occurs behind the viewplane. No yon clipping occurs because it is disabled. These defaults are chosen so as to produce the most straightforward results for a new user.

Why is hither clipping typically enabled? Recall the way the three-dimensional viewing transformation is achieved: projectors passing from an object behind the viewplane to a center of projection in front of it. If some of the object's primitives lie in front of the viewplane, the projectors will not all pass through the viewplane; according to the model, no image should be produced by these primitives. Hither clipping at the viewplane will have exactly this effect. On the other hand, if hither clipping were disabled, unexpected results might be produced. For this reason, hither clipping is usually enabled.

Three-Dimensional Window

JWIND (DWUMIN,DWUMAX,DWVMIN,DWVMAX)	Set window
JCLPW (OPCODE)	Control window clipping

After primitives lying between the depth clipping planes are projected onto the viewplane, they are clipped to the window. The window was discussed previously. It is repeated here to show its relationship to depth clipping and to clarify its role in the three-dimensional viewing transformation.

In two-dimensional applications the window was set in the X-Y world coordinate plane which was coincident with the viewplane. However, using JVDIS and JVPLN, three-dimensional applications establish a viewplane that is, in general, not coincident with the world coordinate X-Y plane. Thus, in the three-dimensional case, the window cannot be defined in world coordinate units.

The placement of the window within the viewplane is relative to the point where a projector from the view reference point pierces the viewplane. For a perspective projection DWUMIN, DWUMAX, DWVMIN, and DWVMAX are specified relative to the point where the projector from the view reference point to the center of projection pierces the viewplane.

For a parallel projection DWUMIN, DWUMAX, DWVMIN, and DWVMAX are specified relative to the point where a parallel projector passing through the view reference point pierces the viewplane.

The effect of locating the window in this manner is that the view reference point will always be visible if DWUMIN and DWVMIN are negative displacements and DWUMAX and DWVMAX are positive. Additionally, if DWUMIN = - DWUMAX and DWVMIN = - DWVMAX, then the view reference point will always appear in the center of the window. Thus, it is extremely easy to center an object in the window-place view reference point on the object and specify the window to be symmetric.

Primitives that are not clipped by the hither or yon planes are subjected to one more clipping operation if window clipping is enabled. They must lie within the window to be displayed. Once the projected primitives have been clipped to the window, the viewing transformation is the same as that for a two-dimensional viewing transformation: the window is mapped to the viewport and the viewport is displayed on the display device.

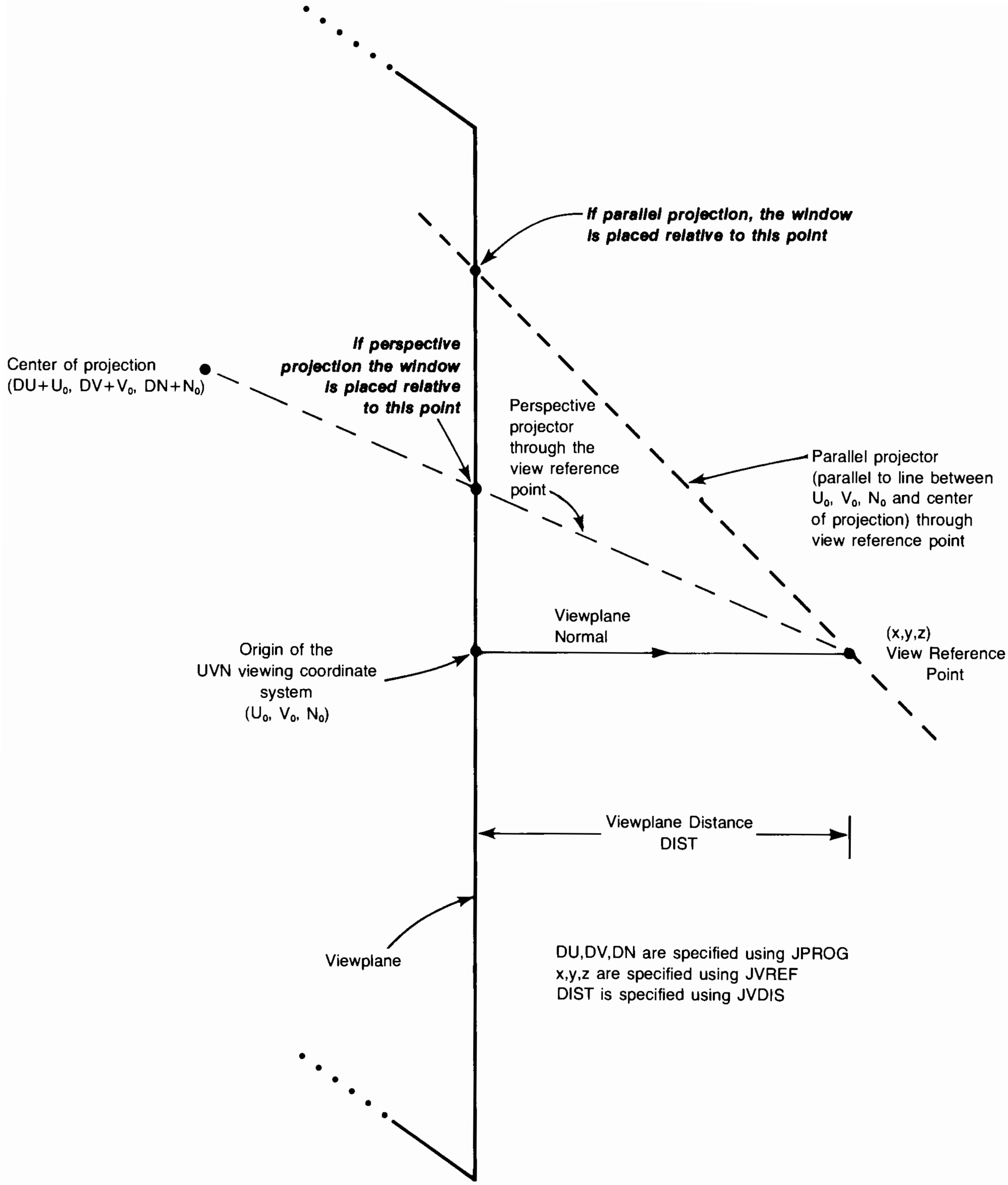

Figure 4-12. Placement of the Window on the Viewplane

Advanced Capabilities

Once the previously described viewing functions are understood, the remaining few should be examined. To see why they are needed, consider the following situation.

A poster entitled Flight is to be created with AGP-3. The principal object in the poster will be an airplane, for which the data is already available. Initially, the problem seems straightforward, however, consider the following:

1. The data for the airplane was created using a right-handed world coordinate system.

2. The data consists of absolute points; that is, it draws the airplane centered around the world coordinate origin. You would like, however, to be able to draw the airplane located anywhere in the world coordinate system.

3. A copy of the airplane only looks good if its wingspan is at least 10% of the width of the viewport. You would like to detect when an airplane's image will be smaller than this.

4. You revise your "size criteria" in (3) to: the airplane only looks good if its wingspan is at least 15 mm. You would like to detect when this is true.

5. Once the airplane has been drawn, you would like to restore the initial viewing transformation so text can be drawn on the poster.

Some of these could be done with a combination of previously described functions; others could not. All would be simply realized with the ability to:

1. set the handedness of the world coordinate system,

2. "translate" an object to some other location,

3. convert world coordinates to virtual coordinates,

4. convert virtual coordinates to millimeters on a particular display device, and

5. reset the viewing transformation to the initial values.

These capabilities are described in the following sections.

Handedness

JHAND (OPCODE) Set handedness of world coordinate system

As defined previously, the world coordinate system is left-handed, meaning the positive Z-axis extends into the page as in Figure 4-13.

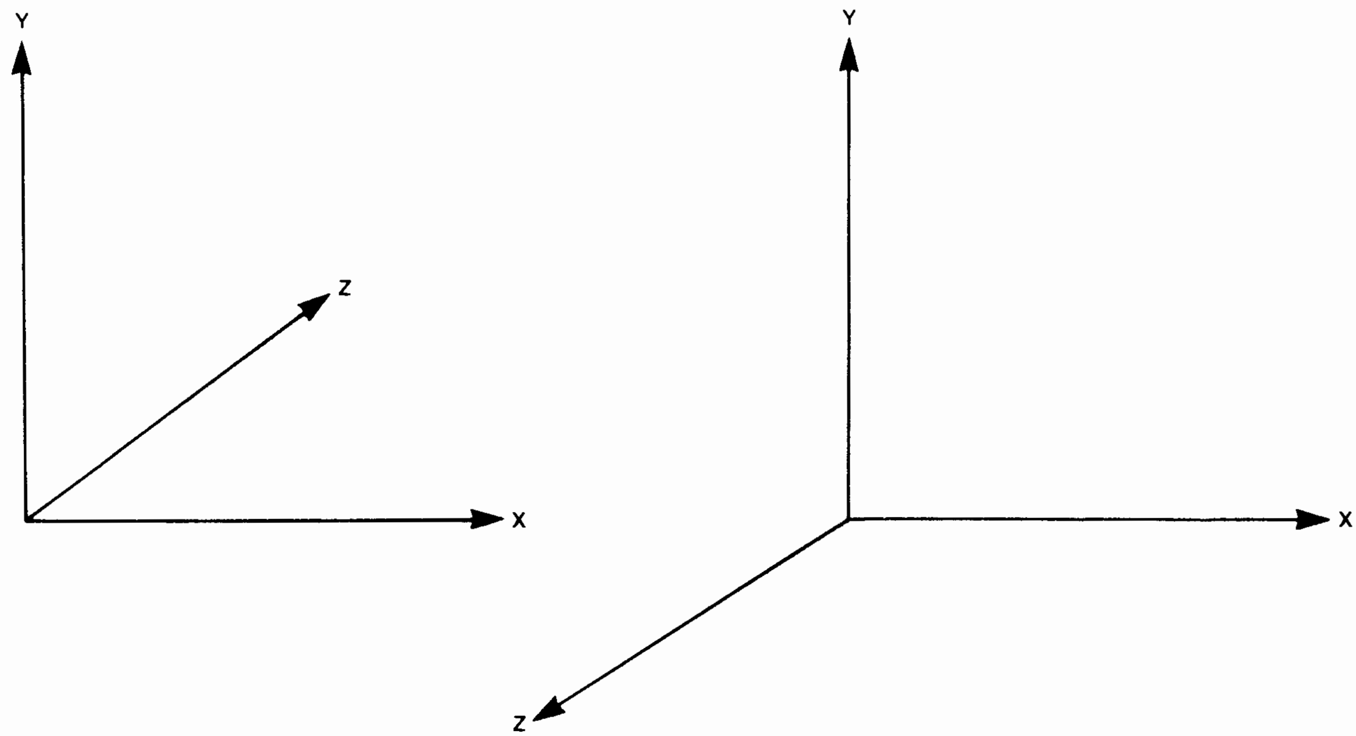

Figure 4-13. Left-handed & Right-handed World Coordinate System.

JHAND may be used to make it left-handed (OPCODE = 0) or right-handed (OPCODE = 1). A right-handed world coordinate system would have the positive Z-axis extending out of the page, toward the user (also shown in Figure 4-13).

It is worth noting that the handedness of the world coordinate system has no effect on the viewing or virtual coordinate systems. The viewing coordinate system is always left-handed. The virtual coordinate system is two-dimensional, hence, handedness is not relevant to it.

Modelling

JDMOD (MATRIX)	Define the 4-by-4 modelling matrix
JCMOD (OPCODE)	Control application of the modelling matrix

Many graphics applications require the ability to manipulate an object before it is viewed. This might consist of scaling (making it bigger or smaller), translating (displacing it as a unit to some other position in the world coordinate system) or rotating (with respect to the world axes). These actions are typically part of a modelling system. A modelling system can be loosely defined as a way of manipulating objects as a whole prior to applying a viewing transformation. In the graphics world, a modelling system is usually thought of as a complete software and hardware package that provides for generation, storage, and output of complex graphical models of the real world. Although AGP-3 can play a very useful role, it is not all that is required in such a system.

The interface between a modelling system and AGP-3 is provided by JDMOD and JCMOD. JDMOD allows the definition of a 4-by-4 matrix which will be applied to primitives when modelling is enabled with JCMOD (OPCODE = 1). This matrix may, for example, scale, translate, or rotate subsequently created primitives before the viewing transformation is applied. Additional software to build and manage the modelling matrix would form another part of the modelling system.

The (row,column) indices of the modelling matrix are assumed to be:

```
(1,1) (1,2) (1,3) (1,4)
(2,1) (2,2) (2,3) (2,4)
(3,1) (3,2) (3,3) (3,4)
(4,1) (4,2) (4,3) (4,4)
```

The following matrices show the role of individual elements of the matrix. Scaling by s:

```
s    0    0    0
0    s    0    0
0    0    s    0
0    0    0    1
```

If, for example, s = 2.0, the result is a two-fold increase in size.

Translation of an object by (m,n,p):

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ m & n & p & 1 \end{matrix}$$

Rotation through the origin of A degrees about the X-axis:

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & \cos A & -\sin A & 0 \\ 0 & \sin A & \cos A & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Rotation through the origin of B degrees about the Y-axis:

$$\begin{matrix} \cos B & 0 & \sin B & 0 \\ 0 & 1 & 0 & 0 \\ -\sin B & 0 & \cos B & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Rotation through the origin of C degrees about the Z-axis:

$$\begin{matrix} \cos C & -\sin C & 0 & 0 \\ \sin C & \cos C & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Combinations of these operations can be performed by matrix multiplications of the above examples. At initialization, the modelling matrix is the identity matrix and modelling is disabled.

Conversion Between World and Virtual Coordinates

```
JVTOW (VX,VY,X,Y,Z)          Convert virtual to world coordinates
JWTOV (X,Y,Z,VX,VY)          Convert world to virtual coordinates
```

The conversion routines JVTOW (virtual to world) and JWTOV (world to virtual) convert a point in one coordinate system to another. For JVTOW the world coordinates corresponding to the virtual point (VX,VY) are calculated using the inverse of the current viewing transformation. The point (X,Y,Z) is always computed to lie in the viewplane. For JWTOV the virtual coordinates corresponding to the world point (X,Y,Z) are returned in (VX,VY). No clipping is performed in this conversion, so (VX,VY) may lie outside of the virtual coordinate rectangle.

Display Device Conversion

```
JDPMM (ID,VX,VY,XMM,YMM)        Convert a virtual point to milli-
                                  meters on the display device
```

JDPMM allows conversion of a virtual coordinate point (VX,VY) to a point in millimeters (XMM,YMM) on the display device of work station ID. Used together with JWTOV, it lets the user determine the size image an object will produce. That is, knowing only the object's world coordinates, the size in millimeters of the resulting image can be programmatically determined. This is very useful in many applications (e.g., those requiring scaled drawings).

In the following program fragment, the length of a line primitive is being determined before it is output. If the X component is less than 1.0 millimeters in length, it is not drawn.

```
  XW1 = -0.5                             * x-coordinate of point 1
  XW2 = 0.5                              * x-coordinate of point 2
  CALL JWTOV (XW1,0.0,0.0,VX1,VY1)       * find virtual point 1 and
  CALL JDPMM (1,VX1,VY1,XMM1,YMM1)       *  convert to mm. on WS 1
  CALL JWTOV (XW2,0.0,0.0,VX2,VY2)       * find virtual point 2 and
  CALL JDPMM (1,VX2,VY2,XMM2,YMM2)       *  convert to mm. on WS 1
  IF (ABS(XMM2-XMM1) .LE. 1.0) GO TO ... * if they are too close...
  CALL J3MOV (XW1,0.0,0.0)               * not too close:
  CALL J3DRW (XW2,0.0,0.0)                  draw the line
```

Resetting the Viewing Transformation

JRSET

Within a program, it is convenient to be able to reset the viewing transformation to a known state in one call. JRSET does this. The viewing transformation is reset to the initial viewing state, with a few differences: handedness, logical display limits, aspect ratio, and viewport are not reset to their initial values.

Table 4-2. Initial Values of Viewing Transformation Components.

Viewing Component	Initial Value
View Reference Point	(0.0,0.0,0.0) {in world coordinates}
Viewplane Distance	(0.0) {between view ref. point and viewplane}
Viewplane Normal	(0.0,0.0,1.0) {relative to viewing coordinate origin}
View Up Vector	(0.0,1.0,0.0) {relative to viewing coordinate origin}
Projection	parallel to (0.0,0.0,-1.0) {perpendicular to the X-Y plane}
Window Clipping	enabled at (-1.0,1.0,-1.0,1.0) {relative to projection of view reference point onto viewplane}
Hither Clipping	enabled at (0.0) {relative to the viewplane}
Yon Clipping	disabled at (0.0) {relative to the viewplane}
Modelling	disabled with matrix set to identity

The handedness of the world coordinate system, the logical display limits currently in effect, and the aspect ratio of the virtual coordinate system remain at their current values. However, the viewport is neither reset to its initial value (a unit square), nor is it left as is, but is reset to the limits of the virtual coordinate system. Recall that this is exactly the effect of a JASPK call.

JRSET returns us (nearly) to the place we began: the initial viewing transformation. It is fitting, then, that it complete the chapter on viewing transformations.

Chapter 5
Segments and Primitives Outside of Segments

General

As graphics applications become more sophisticated, the ability to subdivide complex graphics pictures into logical entities, or segments, becomes valuable. A segment is a named group of output primitives that define an image.

Segments are actually stored in a data structure in the work station program called the segment display area, SDA (see Figure 5-1). The segment display area is simply an in-memory version of the picture on the display device. Without it, once a picture was drawn, the computer would "forget" about it - that is, only the device would have the data with which to recreate the picture. Of course, the application program could redraw the entire picture from scratch. However, that would be a time-consuming process if only one segment were to be modified.

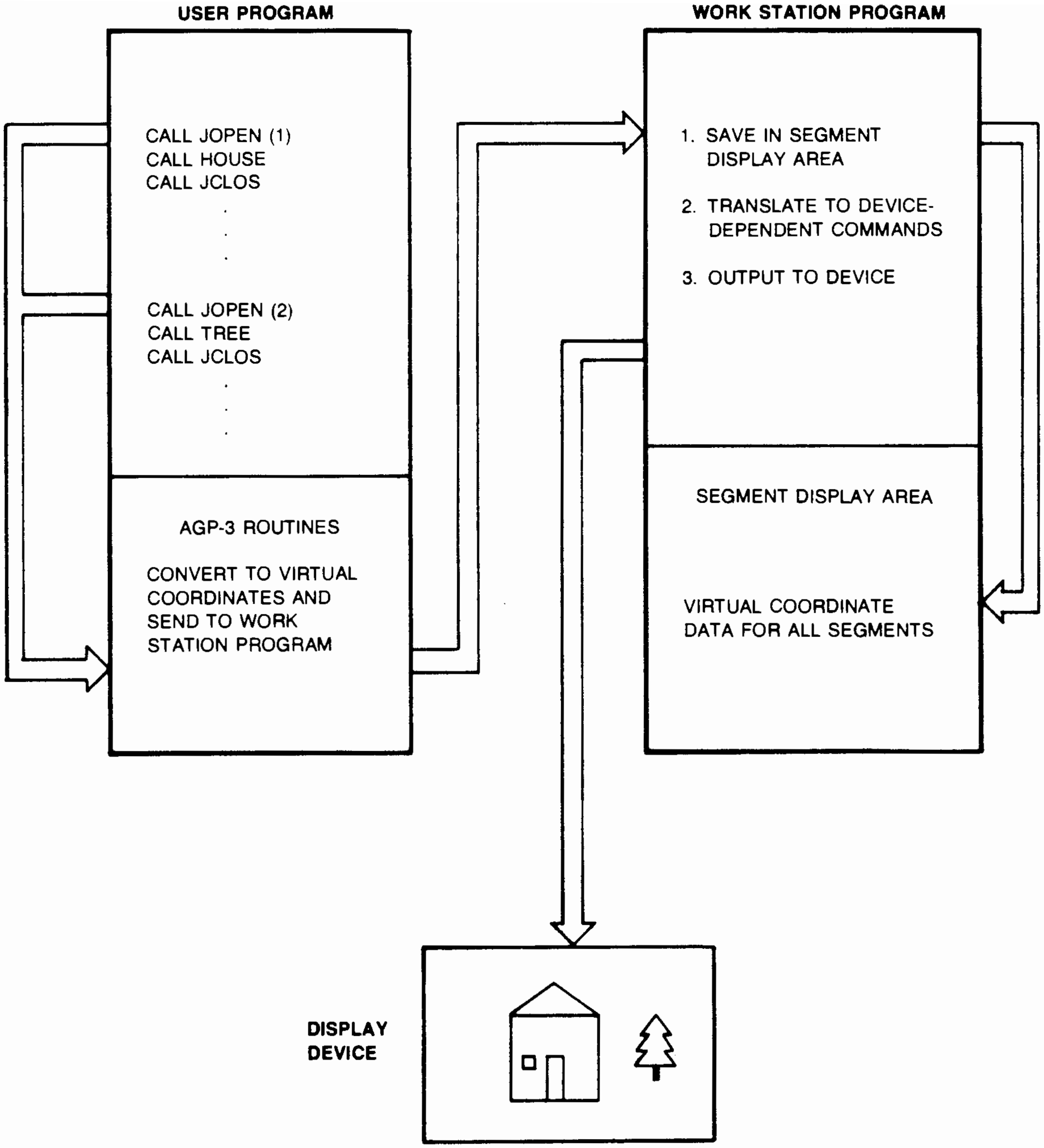

Figure 5-1. Segment Display Area.

Primitives do not have to be included in segments. In some applications, there is no need for the interactivity and higher-level picture modification capability that segmentation provides. The primitives' images could simply be drawn on the display device without saving them in a segment. Indeed, primitives outside of segments have been used in all the examples described previously. Note that primitives outside of segments are temporary in nature. They exist on a display device only until a new-frame-action, which removes all primitives not contained in visible segments.

It is important to recognize that AGP-3 includes no capability for copying a segment into other segments. Also, no segment can reference or call another segment, as in a picture/sub-picture hierarchy. Thus, there is only one level of segmentation for defining a picture.

Segment Creation, Renaming, and Deletion

JOPEN (NAME)	Create/open a segment
JCLOS	Close the segment
JRNAM (ONAME,NNAME)	Rename an existing segment
JPURG (NAME)	Purge an existing segment
JCLR	Purge all existing segments

JOPEN creates and opens a new segment. NAME, an integer, is used to refer to the segment in future calls. The segment is "bound" to each enabled work station; that is, it is saved in the segment display area of each work station being used. JCLOS closes a segment, meaning no more output primitives may be added to it. Once closed, a segment may never be opened again. A segment may be renamed with JRNAM from ONAME to NNAME, and it may be purged from the segment display area and display device of every work station with JPURG. JCLR purges all segments, effectively emptying segment display areas and clearing display devices.

The following program fragment shows a house segment being created, renamed and purged.

```
CALL JOPEN (1)           * open segment 1
CALL HOUSE               * draw a house
CALL JCLOS               * close segment 1
     :
CALL JRNAM (1,23)        * now the house is segment 23
     :
CALL JPURG (23)          * remove the house segment
     :
```

Attributes of Segments

JGHI (OPCODE)	Set system-maintained highlighting attribute
JSHI (NAME,OPCODE)	Set a segment's highlighting
JGDET (OPCODE)	Set system-maintained detectability attribute
JSDET (NAME,OPCODE)	Set a segment's detectability
JGVIS (OPCODE)	Set system-maintained visibility attribute
JSVIS (NAME,OPCODE)	Set a segment's visibility
JVSAL (OPCODE)	Set the visibility of all segments

Segment attributes affect the appearance of the image stored in a segment. Two segment attributes make the segment's image 1) visible or invisible, and 2) highlighted (to draw the operator's attention to it) or non-highlighted. A third segment attribute controls whether the pick input device can be used to select primitives in a segment. This attribute has the values detectable and non-detectable.

Attributes are divided into two classes: output primitive attributes and segment attributes. Output primitive attributes are static; they may not be changed after the primitive is created. An application program can only change an attribute of an output primitive by deleting the segment that contains the primitive, changing the system-maintained attribute value, and then creating a new primitive. This is not necessary for segment attributes. In fact, a primary reason for grouping primitives into segments is so that their segment attributes may be dynamically changed. As with output primitive attributes, system-maintained values for segment attributes are assigned when a segment is created. These system-maintained values are established with the JGxxx routines. During a segment's construction, as well as after it is closed, the values of its attributes may be changed from the original values with the JSxxx routines.

In these routines, OPCODE = 1 turns <u>on</u> an attribute, while OPCODE = 0 turns that attribute off. These are easily remembered as they are the usual significance of binary 1 and 0. NAME is the segment name specified when it was opened. While all but one of these calls treat segments individually, JVSAL sets the visibility attribute of <u>all</u> existing segments, which may be useful to temporarily clear the display device.

The following example shows an invisible house segment being created. After some intermediate processing, the house is made visible.

```
CALL JGVIS (0)             * subsequent segments will be invisible
CALL JOPEN (23)            * create a segment
CALL HOUSE                 *   that contains a house
CALL JCLOS                 * nothing appears on display device
      :
      :
CALL JSVIS (23,1)          * now the house appears
```

Segments are also useful for picking. Each output primitive in a segment has two components that are reported when that primitive is picked. The first is the name of the segment containing the primitive. The second component is the name of the pick-ID when the primitive was created. Pick-ID is an output primitive attribute set by JPKID (see Chapter 3).

Pick-IDs are useful in menus. A menu is a list of options presented to an interactive user, who chooses one of the menu options with a pick device. This menu is a single segment, with each option labeled by a separate pick-ID. When the user selects one of the menu options, the segment name of the menu and the pick-ID of the selected option are returned, as shown below.

```
CALL JOPEN (9)                      * open the menu segment
CALL JPKID (1)                      * create option 1
CALL J2MOV (0.0, 0.8)               *   set current position
CALL JTEXM (12, 12H Jelly Beans)    *   and output text
CALL JPKID (2)                      * create option 2
CALL J2MOV (0.0, 0.6)               *   set current position
CALL JTEXM (12, 12H   Gum Drops)    *   and output text
CALL JPKID (3)                      * create option 3
CALL J2MOV (0.0, 0.4)               *   set current position
CALL JTEXM (12, 12H Candy Canes)    *   and output text
CALL JCLOS                          * close the menu segment
        :
        :
CALL JPICK (1,1,IBUTN,NAME,PICKID)  * issue a pick request
IF (NAME .NE. 9) GO TO 1000         * check for menu segment
IF (PICKID .EQ. 1) GO TO 110        * is it option 1?
IF (PICKID .EQ. 2) GO TO 120        * is it option 2?
IF (PICKID .EQ. 3) GO TO 130        * is it option 3?
        :
```

The pick-ID attribute is particularly useful for menu tables, where each element of the table is to have its own identity, or for picking data points within a data plot.

Segment Display Area Extension

JSDF (ID,NAME,SEC,CRN,SIZE,CONTRL) Create segment display file

JSDF creates a disc file that AGP-3 uses as an extension to the segment display area of the work station program. ID selects which work station that file will be associated with. NAME, SEC, CRN, and SIZE are the name, security code, cartridge and size, respectively, of the file to be created. The file that is defined with the JSDF call should be larger at least by 4 blocks than the segment display area. If it is not, it will not be used. If bit 3 (decimal 8) of the CONTRL parameter is set, the file will be purged if it already exists, then recreated.

If AGP-3 uses up all the memory space it has in the segment display area and no extension is available, all further actions requiring segment display area storage generate errors.

New-Frame-Action

JNEWF Cause a new-frame-action

To erase one of several images on a display device, many devices must be cleared entirely and then redrawn with only the images that are to remain. This is equivalent to erasing part of a chalkboard by erasing the whole thing, then re-writing the part that was not to be erased. Although this is certainly an awkward way to erase a chalkboard, many display devices function in this manner (for example, a plotter), so AGP-3 must treat them appropriately. The mechanism it uses is called a new-frame-action.

A new-frame-action erases all primitives outside of segments from a display device. If no segments have been used at all, it effectively clears the display device. The effect of a new-frame-action depends on the particular device being used. On chart advance plotters the page is advanced before the visible segments are re-drawn. On a terminal, the screen might be cleared then the segments re-drawn, or the terminal may be able to remove primitives outside of segments explicitly without clearing its screen.

JNEWF causes a new-frame-action on every enabled work station. New-frame-actions also occur on individual work stations when:

1. a visible segment is purged (JPURG or JCLR)
2. a visible segment is made invisible (JSVIS)
3. the highlighting of a visible segment is changed (JSHI)
4. the logical display limits are redefined (JDLIM)

Chapter 6
Input

General

Previous chapters have been concerned with the AGP-3 functions that generate output. This chapter is concerned with the opposite process: receiving input.

AGP-3 applications may receive input from the following devices:

1. Button
2. Keyboard
3. Locator
4. Valuator
5. Pick

A button device returns an integer corresponding to a particular button that is pushed, a keyboard returns an alphanumeric string, a locator returns a real coordinate pair, a valuator returns a single real value and a pick device returns a segment name and pick-ID. Refer to Chapter 2 for a review of the nature of these devices.

Two mechanisms exist for obtaining data from input devices: requesting and sampling. Requested input implies that the program will wait until the operator enters a termination command (e.g. presses a key) before continuing. All AGP-3 input devices may be requested. Sampling means no operator response is needed: the current value of the input device is returned without any wait. Only the locator and valuator devices may be sampled; the other input devices are by nature unsuitable for sampling.

Concurrent graphics output to multiple work stations is fundamental to AGP-3. By enabling more than one work station, the same image may be seen simultaneously on several display devices without the application having to create or send it multiple times. This is not the case with input. In general, the responses of an operator must be handled individually by the application. For this reason, each of the calls described in this chapter addresses a specific workstation; that is, each has an ID parameter in which the desired work station is specified.

A single work station may contain, at most, one each of the five logical input devices. More than one logical device may be found in a single physical device. For example, the HP 2648A Graphics Terminal can be used for both keyboard and locator input. Some input devices which are not physically available at a particular work station may be simulated using software and another device. A valuator is simulated on the HP 2648A by the X coordinate (or the Y coordinate) of the graphics cursor.

Most input devices use echoing. Echoing gives operator feedback during an input operation, reflecting its status to allow refinement before it is completed. The characters that appear on a terminal screen when a key is pressed are a familiar example of echoing. In most systems, these characters have actually been echoed by the computer after it has received them from the keyboard. By seeing what is typed, an operator may correct any typographical errors before terminating the input operation by pressing return. With AGP-3, more elaborate echoing than this is possible.

There are two types of echoes: those on the input device itself and those on the graphics display device. All input functions may be echoed on the input device (depending, of course, on the device's echoing capabilities). However, only locator and pick input may be echoed on the display device. These echoes begin at a user-settable point called the locator or pick echo position. Some display device echoes also utilize this point throughout the echo. For example, when a rubber band line echo is used with locator input, the fixed end of the rubber band line begins at the locator echo position (see Figure 6-1).

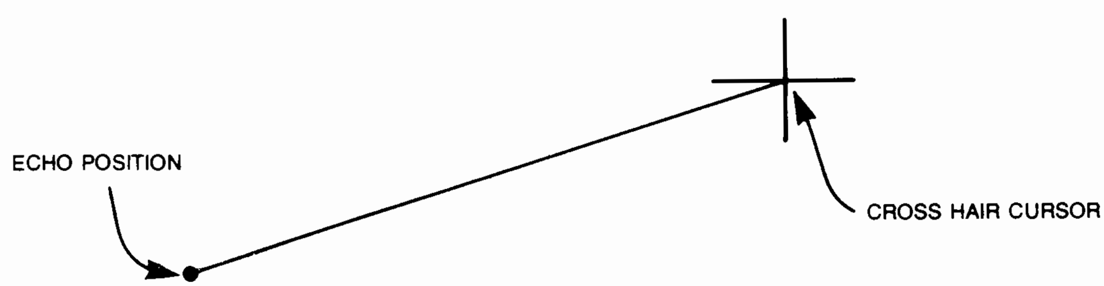

Figure 6-1. Rubber Band Line Echo.

Just as a portion of the display device was selectable for graphics output by setting logical display limits, portions of the locator and pick devices may be selected for input by setting logical locator and pick limits. These limits allow the range of possible operator inputs to be controlled. As shown in Figure 6-2, the logical locator and pick limits map directly to the display surface.

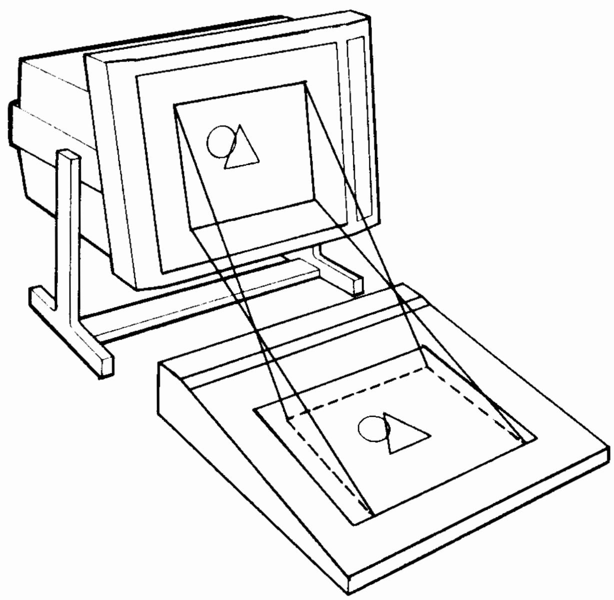

Figure 6-2. Logical Locator or Pick Limits to Logical Display Limits.

Only those points inside the locator limits are eligible to be located, and only those segments within the pick limits are eligible to be picked.

Button

JBUTN (ID,ECHO,VALUE) Await button press, report value

JBUTN requests button input from work station ID. It waits until a button is pressed, then returns the button number in VALUE. If a key other than a valid button key is pressed, 0 is returned.

```
INTEGER VALUE                        * buttons are always integers
        :
        :
        :
CALL JBUTN (1,0,VALUE)               * button input from WS 1, no echo
IF (VALUE .EQ. 1) GO TO 100          * button 1 was pushed
IF (VALUE .EQ. 2) GO TO 200          * button 2 was pushed
IF (VALUE .EQ. 3) GO TO 300          * button 3 was pushed
        :                            * if here bad button was pushed
```

Echoing, as controlled by the ECHO parameter, depends on the capabilities of the input and display device being used. Possibilities include beeping a bell or echoing an integer corresponding to the button that was pressed.

Keyboard

JKYBD (ID,ECHO,MAX,<u>ACTUAL</u>,<u>STRING</u>)   Request text entry, report value

JKYBD requests keyboard input from work station ID. It waits until a line of text has been entered on the keyboard and terminated (e.g., by a carriage return), then returns the text in the array STRING. The characters will be in GRAPHICS/1000-II packed ASCII format (that is, ASCII left-justified, one character per byte and blank filled.). Up to MAX characters will be allowed, and the actual number entered will be returned in ACTUAL.

As with button input, echoing is device dependent.

```
INTEGER ACTUAL, STRING(40)           * 40 integers = 80 characters
        :
        :
CALL JKYBD (1,1,80,ACTUAL,STRING)    * request up to 80 characters
                                     *   from WS 1 with echo 1
CALL JTEXL (ACTUAL,STRING)           * output as low quality text
        :
```

Locator

JWLOC (ID,ECHO,BUTTON,VX,VY)	Request locator input, report value
JSLOC (ID,ECHO,VX,VY)	Sample locator, report value
JLLIM (ID,XMIN,XMAX,YMIN,YMAX)	Set limits on the locator device
JLOCP (ID,PX,PY)	Set the locator echo position
JLPMM (ID,PX,PY,XMM,YMM)	Convert a virtual point to millimeters on the locator device

JWLOC and JSLOC invoke locator input. JWLOC waits for a locator button to be pressed before returning to the application. JSLOC returns the current value of the locator without any wait. Each locator device has its own set of buttons that may or may not be the same buttons that comprise the button device.

The locator returns a virtual coordinate point (VX,VY). The corresponding world coordinate point may be derived with JVTOW, as shown below.

```
  CALL JWLOC (1,1,BUTTON,VX,VY)      * request locator input from
                                     *   work station 1 with echo 1
  CALL JVTOW (VX,VY,X,Y,Z)           * convert to world coordinates
```

Note that the point (X,Y,Z) will always be found in the viewplane. This assumption must be made for JVTOW to be able to derive three coordinates from two.

Depending on the capabilities of the devices being used, locator echoes are possible on either the locator or display device.

JLLIM sets the logical limits of the locator device. The logical locator limits are initially defined as the addressable area of the locator device. With a call to JLLIM, the user can set them to a new rectangle on the locator device. The pairs (XMIN,YMIN) and (XMAX,YMAX) define the corner points of this rectangle in terms of millimeters of offset from the origin of the device. (The position of this origin is device-dependent and is documented in the GRAPHICS/1000-II Device Handler's Manual.)

JLLIM does not affect the virtual coordinate system. It affects only the mapping between the virtual coordinate system and the locator device.

If the locator device and the display device are identical, then the logical locator limits and the logical display limits must be identical. Specifically, when they refer to the same device, the interaction between JDLIM and JLLIM is as follows:

1. The logical locator limits are initialized to the same values as the logical display limits.

2. A call to JDLIM implicitly sets the logical locator limits to the same values.

3. Explicit calls to JLLIM cause an error to be generated. The call is ignored.

The locator echo position is a point in the virtual coordinate system used with locator echoing on the graphics display device. For example, the fixed end of a rubber band line echo will be at the locator echo position. JLOCP sets that position for all locator echoes on the display device. The default locator echo position is in the center of the virtual coordinate system. When a call is made that changes either the viewing transformation or the mapping between the display device and locator device, the locator echo position is reset to the default value. The calls which do this are JASPK, JDLIM, and JLLIM.

JLPMM converts a virtual point to millimeters on the locator device. The virtual coordinate point (PX,PY) is returned in (XMM,YMM) by mapping the virtual coordinate system onto the locator device. The returned values (XMM,YMM) are in millimeters of offset from the origin of the locator device.

The following program section shows a likely use of these functions. Locator input is used to let the operator set the logical locator limits interactively by selecting first the lower left corner then the upper right corner.

```
CALL JWLOC (1,2,BUTTON,VXMIN,VYMIN)     * request locator from WS 1
                                        *   with display device echo
CALL JLOCP (1,VXMIN,VYMIN)              * set it as echo position
CALL JWLOC (1,4,BUTTON,VXMAX,VYMAX)     * request locator input from
                                        *   WS 1, rubber band echo
CALL JLPMM (1,VXMIN,VYMIN,XMMIN,YMMIN)  * convert both points to
CALL JLPMM (1,VXMAX,VYMAX,XMMAX,YMMAX)  *   millimeters on locator
CALL JLLIM (1,XMMIN,XMMAX,YMMIN,YMMAX)  * set locator limits to them
```

Valuator

JWVAL (ID,ECHO,SUBVAL,BUTTON,VALUE)	Request valuator input, report value
JSVAL (ID,ECHO,SUBVAL,VALUE)	Sample valuator, report value

JWVAL and JSVAL invoke valuator input. JWVAL waits for the valuator button to be pressed before returning. JSVAL returns valuator information without any wait. Each valuator device has its own set of buttons (used only in the valuator request) that may or may not be the same as those that comprise the button device.

A valuator device has multiple valuators, called subvaluators, on it. For example, the HP 2648A Graphics Terminal may use either the X coordinate or the Y coordinate of the graphics cursor as a valuator. SUBVAL is used to select a particular valuator to be used. Recall that a valuator device returns a real value between 0.0 and 1.0. If SUBVAL = 1, the X position of the HP 2648A's cursor is returned, with 0.0 represented as the extreme left side of the display and 1.0 represented as the extreme right side. If SUBVAL = 2, the Y position of the cursor is returned, with 0.0 represented as the bottom of the display and 1.0 represented as the top of the display.

Echoing is possible with both JWVAL and JSVAL. The type and number of echoes available is device dependent. Most valuator devices support at least one form of echoing such as beeping, displaying the value sampled, etc.

Pick

JPICK (ID,ECHO,BUTTON,NAME,PICKID)	Request button input, return segment name and pick-ID
JPLIM (ID,XMIN,XMAX,YMIN,YMAX)	Set limits of the pick device
JPIKP (ID,PX,PY)	Set the pick echo position
JPPMM (ID,PX,PY,XMM,YMM)	Convert a virtual point to millimeters on the pick device

JPICK requests pick input, returning the name of the selected segment and name of the pick-ID attached to the output primitive in NAME and PICKID. For the operator, the pick operation consists of moving a cursor or light pen near a primitive, then pressing a pick button. AGP-3 searches the segment display area for a detectable, visible segment and primitive that is near the location chosen. If no segment can be identified, NAME and PICKID will be 0. For a complete description of the pick algorithm used, see JPICK in the AGP-3 Reference Manual.

When a primitive is created within a segment, the current pick-ID is bound to it. Recall that JPKID sets that pick-ID attribute. Primitives outside of a segment are not stored in the segment display area and therefore cannot be picked.

Each pick device has its own set of buttons. These may or may not be the same set of buttons that make up the button device.

Pick input may be echoed either on the graphics display device or on the pick device itself. Both devices must support the echo for it to be available.

The following example was presented previously when segments were described. It shows picking being used to select choices in a menu.

```
CALL JOPEN (1)                        * open the menu segment
CALL JPKID (1)                        * create choice 1
CALL J2MOV (0.0, 0.8)                 *   set current position
CALL JTEXM (12,12H Jelly Beans)       *   and output text
CALL JPKID (2)                        * create choice 2
CALL J2MOV (0.0, 0.6)                 *   set current position
CALL JWTOV (0.0, 0.6, 0.0, VX, VY)    * translate this point to
                                      *   virtual coordinates
CALL JPIKP (1, VX, VY)                * set pick echo position
CALL JTEXM (12,12H   Gum Drops)       * output text for choice 2
CALL JPKID (3)                        * create choice 3
CALL J2MOV (0.0, 0.4)                 *   set current position
CALL JTEXM (12,12H Candy Canes)       *   and output text
CALL JCLOS                            * close the menu segment
        :
        :
CALL JPICK (1,1,IBUTN,NAME,PICKID)    * issue a pick request
IF (NAME .NE. 1) GO TO 1000           * check for menu segment
IF (PICKID .EQ. 1) GO TO 110          * choice 1
IF (PICKID .EQ. 2) GO TO 120          * choice 2
IF (PICKID .EQ. 3) GO TO 130          * choice 3
                                      * bad choice if here
```

JPLIM specifies the logical limits of the pick device. The logical limits are initially defined as the addressable area of the pick device. With a call to JPLIM, the user can set them to a new rectangle on the pick device. The pairs (XMIN,YMIN) and (XMAX,YMAX) define the corner points of this rectangle in terms of millimeters of offset from the origin of the device.

JPLIM does not affect the virtual coordinate system. It affects only the mapping between the virtual coordinate system and the pick device.

If the pick and the graphics display are both the same physical device, then the logical pick limits and the logical display limits must be identical. Specifically, the interaction between JDLIM and JPLIM when they refer to the same device is as follows:

1. The logical pick limits are initialized to the same values as the logical display limits.
2. A call to JDLIM implicitly sets the logical pick limits to the same values.
3. Explicit calls to JPLIM are ignored.

For all pick echoes on the graphics display device, the echo originally appears at the pick echo position, a point in the virtual coordinate system set by JPIKP. Initially, it is centered in the virtual coordinate system, and is reset to there by any call that changes either the viewing transformation or the mapping between the display device and pick device. Those calls are JASPK, JDLIM, and JPLIM.

A conversion routine, JLPMM, is provided to convert virtual coordinates to millimeters on the pick device. The virtual coordinate point (PX,PY) is returned in (XMM,YMM) by mapping the virtual coordinate system onto the pick device. The returned values (XMM,YMM) are in millimeters of offset from the origin of the pick device.

Chapter 7
Control

General

This chapter completes the description of AGP-3, with a discussion of the control functions. Some control functions initialize and modify the graphics environment. Others interactively inquire the environment's features - for example, the availability of a button input device. Still others control the timing of image generation. Beyond these general categories, escape functions and error control are also included.

Initialization and Termination

The following functions control the AGP-3 system, work stations, and devices in those work stations.

AGP-3 System Initialization and Termination

JBEGN	Begin AGP-3
JEND	End AGP-3

JBEGN is used to initialize the graphics system. It should be used before any other AGP-3 call is made. Likewise, after the last AGP-3 call, JEND should be called to terminate the graphics system. JBEGN guarantees a standard state in which all attributes and viewing transformation components are at their default values.

```
PROGRAM FOO
CALL JBEGN        * from this point on AGP-3 calls may be made
     :
     :
     :
CALL JEND         * all done with AGP-3
END               *   and with program FOO
```

In the course of terminating AGP-3, JEND may implicitly call other control functions that the user program had failed to specify itself.

Work Station Initialization and Termination

JDINT (ID,PGM,LU,CONTRL)	Initialize a work station
JWEND (ID)	Terminate a work station

Before a work station may be accessed in any way, it must be initialized with JDINT. JWEND terminates a work station and the work station program associated with it. Up to eight work stations may be initialized concurrently.

The parameters to JDINT have the following significance. A work station is identified by the ID parameter, an integer between 1 and 8. For example, if three work stations are being used, they are typically referred to as work station 1, work station 2, and work station 3. However, if desired, they could be designated work stations 6, 2, and 5. JDINT associates a work station ID with the work station program whose name is in the PGM parameter. The LU of the graphics display device is passed in the LU parameter. (In a session environment, the LU must be in the operator's Session Switch Table.) Upon being initialized, the graphics display of the work station is cleared.

The last parameter in JDINT is CONTRL, the control word. It selects certain characteristics of the operating system to work station interface. These include whether system spooling of the LU will be possible, whether the LU will be locked while the work station is initialized, and whether simulated raster erase will be used on the display device. (See the AGP-3 Reference Manual for details.)

All device-dependent primitive attribute tables are initialized with JDINT.

Enabling and Disabling Work Stations

JWON (ID)	Turn a work station on for output
JWOFF (ID)	Turn a work station off for output

JWON enables a work station to accept subsequent graphics output. Work stations are originally disabled, and may not be enabled until JDINT has initialized them. Each display device will respond in its own way when first enabled. For example, a plotter may pick up a pen. Subsequently created output primitives will be sent to it, and new segments will be saved in its segment display area. JWOFF removes a work station from the enabled list and initiates some device-dependent actions. For example, on a plotter, it causes the pen to be put away. However, any primitives already on a display device will be left as is.

The interaction between the display device and segment display area of a disabled work station is a complex one. When disabled, the work station's display device is no longer updated. New segments are not stored in its segment display area, but changes to old segments are. If a segment that was bound to that work station before it was disabled is purged, or its visibility or highlighting is changed, the contents of the segment display area will reflect it. If the work station is once again enabled, the contents of the segment display area and the display device will not be the same. This inconsistency may only be remedied by JNEWF (or a call that does a new-frame-action implicitly). These circumstances are demonstrated below:

```
CALL JDINT (1,...)     * initialize work station 1
CALL JWON (1)          * enable work station 1 for output
    :
    :
CALL JOPEN (10)        * create segment 10
{output primitives}    *   and fill with primitives
CALL JCLOS             *   then close it
    :
    :
CALL JOPEN (12)        * create segment 12
{output primitives}    *   and fill with primitives
CALL JCLOS             *   then close it
    :
    :
CALL JWOFF (1)         * disable work station 1 for output
                       *   (segments 10 and 12 are on the display)
CALL JPURG (10)        * segment 10 is deleted but remains on work
                       *   station 1's display because it is disabled
CALL JWON(1)           * segment 10 still remains ....
CALL JNEWF             * segment 10 is removed from the display now
```

Enabling and Disabling Devices

JEDEV (ID,CLASS,LU)	Enable a logical device
JDDEV (ID,CLASS)	Disable a logical device

JEDEV and JDDEV enable and disable devices at work station ID. The type of device is specified with the CLASS parameter. Possible devices are the alphanumeric display and all input devices: button, keyboard, locator, valuator, and pick. Values of the CLASS parameter are:

```
CLASS = 1 - Alphanumeric device
      = 2 - Button device
      = 3 - Keyboard device
      = 4 - Locator device
      = 5 - Valuator device
      = 6 - Pick device
```

As a rule, a device must be enabled before it is used. Also, the work station on which it is being enabled must have been previously initialized. For example,

```
CALL JDINT (2,6HWSP   ,1,0)    * initialize a work station program,
      :                        *   name='WSP', LU=1, no control bits
      :
CALL JEDEV (2,4,1)             * enable work station 2's locator
CALL JWLOC (2,...)             *   device at LU 1 then use it
      :
```

Note that a device is enabled with JEDEV independently of graphics output being enabled through JWON.

An input device will not be enabled if LU is the same LU as a display device being outspooled. AGP-3 will ignore a JEDEV that mentions an LU previously assigned to spooled output and give an error.

Not all work station programs will be configured to have a logical device of each type; in fact, it is possible to have a work station with only one device. The devices to be included in a work station program are determined at load time. If JEDEV attempts to enable a logical device that the work station program does not have, an error will be generated. To prevent this, the available devices may be inquired any time after a work station is initialized.

The following example demonstrates initialization and termination of a work station, as well as enabling an input device.

```
CALL JDINT (1, 6HWSP   ,LU,0)  * initialize work station 1 and
CALL JWON (1)                  *   turn it on for graphics output
       :
       :
CALL JEDEV (1,4,LU)            * enable the locator,
CALL JWLOC (1,0,BUTTON,VX,VY)  *   use it,
CALL JDDEV (1,1)               *   and end it
       :
CALL HOUSE                     * send graphics output
       :
       :
CALL JWOFF (1)                 * turn off graphics output
CALL JWEND (1)                 *   and terminate the work station
```

Inquiry

A variety of inquiry functions are available. The status of the picture generation process, the status of AGP-3, and the physical properties of a work station are general classes of information available through the several inquiry functions.

Work Station Inquiry

JIWS (ID,OPCODE,ISIZE,RSIZE,<u>ILIST</u>,<u>RLIST</u>)   Inquire work station traits

Work station inquiry is very useful. What is the aspect ratio of the display device? How many linestyles does it have? Questions like these may be answered by JIWS. As in previous calls, ID selects the work station to be queried. OPCODE specifies what particular information is desired. Answers are returned in two arrays: ILIST if they are integers or RLIST if they are reals. These arrays must be sized appropriately and their sizes passed in ISIZE and RSIZE.

There is a correspondence between the values of OPCODE and the type of information returned. The rule is as follows: in an opcode, the hundreds digit indicates the number of reals to be returned, while the thousands digit indicates the number of integers. The remaining digits (the tens and ones) are simply used to make each opcode unique. As an example, two OPCODEs are:

```
 254    Aspect ratio of the graphics display device
        RLIST(1) = Aspect ratio of virtual coordinate system
             (2) = Aspect ratio of logical display limits


5050    Graphics display device information
        ILIST(1)-ILIST(3) = Six character device name
        ILIST(4) = Status
           = -1  there is no graphics display device
           =  0  not enabled for graphics output
           =  1  enabled for graphics output
        ILIST(5) = LU of graphics display device (0 if
                   no graphics display device enabled)
```

As may be seen, opcode 254, with a 2 in the hundreds digit, returns two reals, while 5050, with a 5 in the thousands digit, returns five integers. The remaining JIWS opcodes are documented under the JIWS routine in the AGP-3 Reference Manual.

One use of JIWS is device optimization: the use of inquiry to enhance the application's utilization of devices available to it. For example, it may inquire whether a pick input device is currently available. If so, the application will continue with the preferred pick algorithm. If not, it may use a keyboard entry algorithm instead. In this way, the interactive capabilities of the environment are fully utilized while preserving device independence.

The following example shows how JIWS can be used to set the virtual coordinate system's aspect ratio to that of a work station's logical display surface, making the entire display available in a device-independent manner.

```
INTEGER ILIST(1)                    * 1 integer of information
REAL RLIST(2)                       * 2 real numbers of information
      :
CALL JDINT (1,...)                  * initialize work station 1
CALL JIWS  (1,254,0,2,ILIST,RLIST)  * inquire its logical display
                                    *   limits
CALL JASPK (1.0,RLIST(2))           * set aspect ratio accordingly
      :
```

Output Primitive Inquiry

JICP (X,Y,Z)	Returns current position
JITSZ (NCHARS,STRING,SHIGH,SWIDE)	Returns text height and width

Graphics output primitives begin at the current position, inquirable with JICP. It returns a world coordinate triple, (X,Y,Z), corresponding to the current position in world coordinate space.

JITSZ determines the height and width of a high quality text string before it is output. NCHARS is the number of characters in STRING. The height and width returned in SHIGH and SWIDE are calculated in world coordinates using the current text attributes: size, gap, and font. Note that the character base vector, the character plane vector, slant, and justification do not affect the height or width of a high quality text string.

The following subroutine uses text size inquiry to draw a string in a user-specified rectangle. Its design is complicated by the fact that high quality text fonts may be of variable width. The subroutine accomodates these by setting character size in two steps: first the size is approximated as if the font were fixed width, then the string width this size produces is inquired and used to refine the character size. Once the character size is set, the string is output as high quality text.

```
SUBROUTINE FIT (NCHARS, STRING, XMIN, XMAX, YMIN, YMAX)
INTEGER NCHARS, STRING (80)                  * draws a text string in
REAL XMIN, XMAX, YMIN, YMAX                  *   the given dimensions

REAL WIDTH, HEIGHT, SHIGH, SWIDE

WIDTH  = (XMAX - XMIN) / COUNT               * set character height
HEIGHT = YMAX - YMIN                         *   and width as if the
CALL JCSIZ (WIDTH, HEIGHT, 0.0)              *   font were monospaced
CALL JITSZ (NCHARS, STRING, SHIGH, SWIDE)    * use width of the string
WIDTH = WIDTH / (SWIDE / NCHARS)             *   this size produces to
CALL JCSIZ (WIDTH, HEIGHT, 0.0)              *   calculate final size

CALL J2MOV (XMIN, YMIN)                      * move absolute
CALL JJUST (0.0, 0.0)                        * set justification
CALL JTEXH (NCHARS, STRING)                  * draw high quality text

RETURN
END
```

Viewing Transformation Inquiry

```
JIWND (XARRAY,YARRAY,ZARRAY)      Inquire window in world coordinates
JIMAT (OPCODE,MATRIX)             Inquire internal matrices
```

JIWND and JIMAT inquire features of the viewing transformation. JIWND returns the four corner points of the window in world coordinates as ordered triples in XARRAY, YARRAY, and ZARRAY. This is the window set previously by JWIND. However, recall that the parameters to JWIND were in viewing coordinates, not world coordinates. JIWND does the conversion from viewing to world coordinates. The following program shows how JIWND can be used to draw a rectangle that corresponds to the window.

```
REAL XARRAY(4),YARRAY(4),ZARRAY(4)         * 4 points to be returned
     :
     :
CALL JIWND (XARRAY,YARRAY,ZARRAY)          * get window's corners
                                           *   in world coordinates
CALL J3PLY (4,XARRAY,YARRAY,ZARRAY)        * draw 3 sides of window
CALL J3DRW (XARRAY(1),YARRAY(1),ZARRAY(1)) * now draw side 4
     :
```

There are three internally maintained matrices accessible with JIMAT. The first is the viewing transformation. Set by the parameters of the viewing functions, it is used to translate from world coordinates to virtual. The second, known as the inverse viewing transformation, is simply the inverse of the first. It converts from virtual coordinates back to world. Finally, the third, known as the modelling transformation, is the user-defined modelling matrix set directly by JDMOD. The matrix to be returned is determined by OPCODE:

1 - viewing transformation
2 - inverse viewing transformation
3 - modelling transformation

One need for inquiry is modification. A modelling user might like to incrementally change his modelling transformation by inquiring it, changing one or two of its elements, then establishing the updated one with JDMOD. Inquiry of the modelling matrix facilitates its later modification.

Segment Inquiry

```
JISGA (NAME,VALUE)                      Returns segment attributes
JISGW (SEGNAM,MAXNAM,COUNT,IDARRY)      Returns names of work stations
                                             a segment is bound to
```

Segment inquiry is available through JISGA and JISGW. JISGA returns the attributes of a segment, while JISGW returns the names of work stations it is bound to. The three segment attributes are highlighting, visibility and detectability. Recall that they are dynamically modifiable (through JSHI, JSVIS, and JSDET respectively) and hence inquiry is a very useful tool with which to control them.

JISGA can be used to find the current attribute values of the segment identified by NAME. The attributes are returned in the VALUE array according to the following table:

```
VALUE(1) = -1 - the segment doesn't exist
            0 - the segment is not highlighted
            1 - the segment is highlighted

VALUE(2) = -1 - the segment doesn't exist
            0 - the segment is not visible
            1 - the segment is visible

VALUE(3) = -1 - the segment doesn't exist
            0 - the segment is not detectable
            1 - the segment is detectable
```

Notice that this call can be used to determine whether or not a segment exists: all three VALUEs will be -1 if it does not.

The following example shows JISGA being used when toggling the state of the visibility attribute.

```
INTEGER NAME, VALUE(3)            * segment name and attribute array
      :
      :
NAME = 9                          * segment 9 is of interest
CALL JISGA (NAME,VALUE)           * find its visibility in VALUE(3):
IF (VALUE(2) .EQ. -1) GO TO ...   *   doesn't exist => can't change
IF (VALUE(2) .EQ. 0) OPCODE = 1   *   is not visible => make visible
IF (VALUE(2) .EQ. 1) OPCODE = 0   *   is visible => make not visible
CALL JSVIS (NAME,OPCODE)          * reset the visibility attribute
      :
```

JISGW finds what work stations a segment is bound to. To be bound to a work station means to exist in its segment display area. The number of work stations containing the segment in question is returned in COUNT, and their names are placed in the integer array IDARRY (MAXNAM is the size of IDARRY). A segment does not have to appear on the display device to be returned by JISGW. This would happen, for example, if the visibility attribute of that segment were set to invisible.

Real and Integer Inquiry

JI1IN (OPCODE,VALUE)	Returns integer information
JI1RE (OPCODE,VALUE)	Returns 1 real value
JI2RE (OPCODE,VAL1,VAL2)	Returns 2 real values
JI3RE (OPCODE,VAL1,VAL2,VAL3)	Returns 3 real values
JI4RE (OPCODE,VAL1,VAL2,VAL3,VAL4)	Returns 4 real values

The remaining inquiry functions can best be described as "miscellaneous". They vary by type (integer or real) and quantity (1,2,3 or 4) of the values returned in VALUE. OPCODE is always used to select which particular question is being asked. Its values and the information inquirable are documented under these calls in the AGP-3 Reference Manual.

Timing

Consider the following identifiable components of the AGP-3 system:

1. operator
2. user program
3. work station program
4. graphics device

The operator interacts with a user program, which transmits device-independent commands to the work station program, where they are translated into device-dependent commands and sent to the graphics device. To produce computer graphics, each of these components must work together. A primary feature of this interaction is timing: when is each component "doing something"? This section is concerned with understanding and controlling timing.

Note that AGP-3's timing choices are unrelated to operating system buffering. Instead, they are concerned only with the buffering done within the user program and between the work station program and the user program. AGP-3 has no control over buffering in the operating system.

For purposes of simplification, the following discussion assumes only one work station is initialized and enabled. To comprehend the different timing modes possible, assume that the simplest one, called immediate visibility, is in use. In immediate visibility, each output primitive is transmitted to the device prior to returning from the output call. This timing mode is least efficient in utilization of the system's resources. Figure 7-1 illustrates immediate visibility.

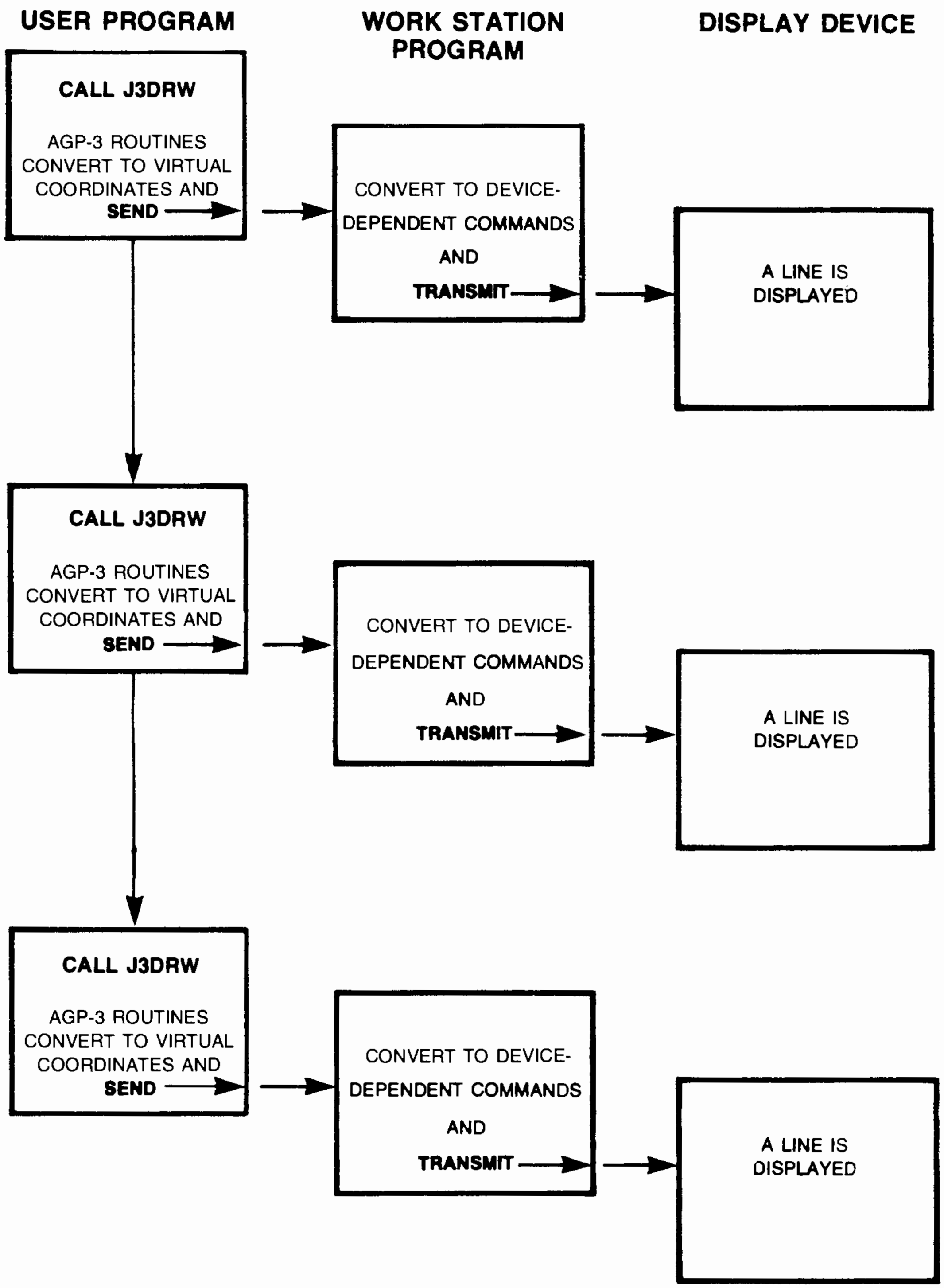

Figure 7-1. Immediate Visibility.

In system buffering mode, a number of output primitives are saved together in the user program before transmission to the work station program. In turn, the work station program buffers commands to the display device, as shown in Figure 7-2. An output primitive is not necessarily transmitted to the device prior to returning from the output call. With system buffering, the use of system resources for communication between the user and work station programs, as well as between the work station program and the display device, is optimized.

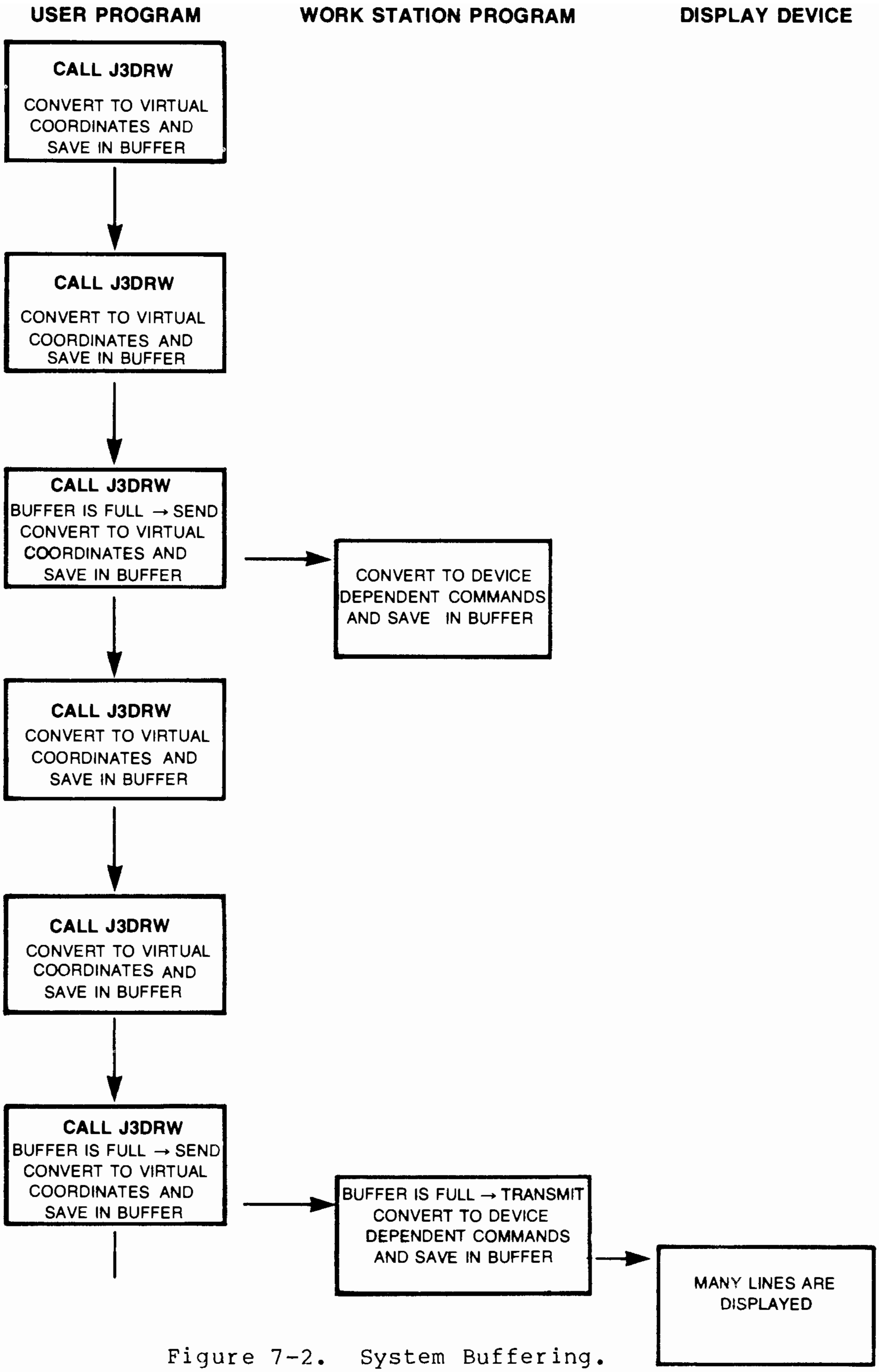

Figure 7-2. System Buffering.

Another timing idea is "batching". Batching can be roughly defined as saving everything up for an interval of time, and then sending it all at once. In AGP-3, this is called batch of updates. To update the display device is to add an output primitive, remove a segment or change a segment's attribute. A batch of updates is the accumulation of display device updates for execution as a group.

Batch of updates is the opposite of immediate visibility. When the program requests batch of updates, not only are updates not sent to the display device individually, but they are not sent at all until the batch of updates is ended. Instead, they are saved in the segment display area. Notice that even primitives outside of segments are temporarily saved in the segment display area during a batch of updates.

It must be emphasized that batching and buffering only affect when an image reaches a display device. They have no affect whatsoever on what the final picture looks like. With this in mind, let us examine these functions further.

Immediate Visibility

JIVON	Set batching mode to immediate visibility
JIVOF	End immediate visibility batching mode

When JIVON has been used to request immediate visibility, display device updates are sent to enabled work stations before the subroutine is exited. Immediate visibility is cancelled with JIVOF, which causes return to the default mode (system buffering).

Immediate visibility is the least efficient of the batching modes. It should be used only when display device updates must take place as soon as they are defined, even if the finished picture will take longer to create. JMCUR is an alternative to immediate visibility that may be used to ensure that the display device is fully updated when in system buffering mode.

System Buffering

JMCUR	Make all enabled display devices current

System buffering is the default batching mode of AGP-3. When it is in effect, display device updates requested by an AGP-3 function are not necessarily sent to work stations before return from the function. When in system buffering mode, the display device may be made current at any time with JMCUR, which forces all currently buffered updates to be sent immediately. Note that JMCUR does not change the timing mode from system buffering.

Many AGP-3 functions make the display device current automatically. Each of the input routines, namely JBUTN, JKYBD, JWLOC, JSLOC, JWVAL, JSVAL, and JPICK, make it current to ensure the operator has received all relevant updates before input is requested. Additionally, enabling a work station (JWON), disabling a workstation (JWOFF), changing timing modes (JIVON,JBATC, and JUPDT), and error inquiry and setting (not discussed yet, but done with JIERR and JSERR) each implicitly update the display device. There is no need for a program to explicitly call JMCUR before any of these functions.

Batch of Updates

JBATC	Begin batch of updates
JUPDT	End batch of updates

JBATC sets the batching mode to batch of updates and JUPDT ends the batch of updates. Display device changes requested inside a batch of updates - that is, between calls to JBATC and JUPDT - will not appear on the display device until JUPDT is called. Instead, they are saved in the segment display area of all enabled work stations. Not until the batch of updates is ended do the changes appear on the display devices of enabled work stations. No alphanumeric output calls (JALPH), input calls (JBUTN, JWVAL, JPICK, etc.) or initialization control calls (JDINT, JWON, JWOFF, JEDEV, JDDEV, JWEND) should be made within a batch of updates. An attempt to do so will result in an error and the batch of updates being closed through an implicit call to JUPDT.

When a segment is purged from a work station, AGP-3 deletes the segment from its segment display area, then implicitly causes a new-frame-action to occur. Typically a new-frame-action erases the display device and then re-sends the contents of the segment display area. This is an efficient technique when only one segment is being purged. However, what about when two or more segments are purged sequentially? Multiple new-frame-actions are generated, typically causing the display device to be re-drawn many times with only one segment removed at a time. Batch of updates will prevent this, allowing only one new-frame-action after all deletions have been made from the segment display area.

The following example illustrates this idea:

```
CALL JBATC          * hold off subsequent display device changes
CALL JPURG (1)      * purge segment 1
CALL JPURG (2)      *   and segment 2 ...
     :
     :
CALL JPURG (n)      *   through segment n
CALL JUPDT          * update the display device now
```

Without the use of batch of updates, the purge sequence would implicitly cause 'n' new-frame-actions, which would typically re-draw the screen 'n' times. If only the final result is of interest, in the intermediate steps the screen is re-drawn needlessly. Using JBATC and JUPDT guarantees that at most one new-frame-action is generated. Note that the final picture produced will be the same whether or not batch of updates is used - only the intermediate steps differ.

Another example shows a second use of batch of updates. The following program section creates a series of rotated views of an object. The subroutine SVIEW modifies the viewing transformation in effect, while HOUSE plots the object as a series of primitives outside of segments. Batch of updates lets the next view of the object be computed and stored temporarily in the segment display area before deleting the previous view from the display device.

```
DO 5 DEG = 1,360,10   * rotate 360 degrees in 10 degree increments
   CALL JBATC         * begin batch of updates (so there will
                      *   be no display changes until JUPDT)
   CALL JNEWF         * clear old primitives left from last loop
   CALL SVIEW (DEG)   * set up rotation
   CALL HOUSE         * plot object as primitives outside segments
   CALL JUPDT         * end batch, causing old picture to be removed
5  CONTINUE           *   (due to JNEWF) and new one to replace it
```

The old view of the object is left on the display device until computation of the new one is complete. In this way, computation of the next view can continue while the current view is being viewed.

Escape Functions

JOESC (ID,OPCODE,ISIZE,RSIZE,ILIST,RLIST) Output escape access
JIESC (ID,OPCODE,ISIZE,RSIZE,ILIST,RLIST) Inquiry escape access

While AGP-3 provides a device-independent method of addressing graphics devices, it does recognize that some devices may have capabilities not directly supported. For example, the HP 2648A Graphics Terminal has "area fill", meaning a rectangular area may be filled with a user-definable pattern. This feature is not accessible through the usual output primitives of AGP-3, but is still a useful one in many applications. Therefore, a convenient, standard mechanism is provided for addressing such device-dependent features.

This standard mechanism is called an escape function. Escape functions are of two types: output escape functions (sent by JOESC) provide access to the graphics display device, and inquiry escape functions (sent by JIESC) return information about the device. The particular feature to be accessed is chosen via a unique integer, OPCODE. Since the different functions OPCODE may select are device-dependent, its possible values are documented in the GRAPHICS/1000-II Device Handler's Manual. If the OPCODE used is not supported on a particular device, an error will be produced and the call will be ignored.

Escape functions are device-dependent. Remembering that one of the goals of an AGP-3 application program is to remain device-independent, escape functions can threaten that objective. If the particular hardware feature being used is not available on other devices, then, if possible, a software substitute should be provided. Also, escape functions should be localized to make subsequent modification as easy as possible.

Escape functions may be used anywhere in an application program, with certain restrictions. JOESC and JIESC should not be called during a batch of updates or while a segment is open. Inquiry escape functions should not be directed to a work station that is being outspooled. If any of these conditions are violated, an error will be generated and the call will be ignored.

The ISIZE, RSIZE, ILIST and RLIST parameters concern integer and real arrays whose significance depends on the opcode chosen. ISIZE is the size of the integer array ILIST, while RSIZE is the size of real array RLIST. If ISIZE or RSIZE do not equal the required number of parameters, an error will be produced and the call will be ignored.

AGP-3 strives at all times to know exactly what is occurring on each of the work stations in use. The segment display area allows AGP-3 to maintain its own version of the display device. Escape functions allow the application programmer to circumvent the AGP-3 subsystem and interact with devices directly. Although AGP-3 maintains the escape function avenue, it does not keep track of the use made of it. Output escape functions are not saved in the segment display area. Also, viewing transformations are not applied to them. In fact, escape functions may threaten the integrity of the work station interface, causing AGP-3 to "lose track" of the state of the work station. The result, from the application program's point of view, is that escape functions should be used carefully.

Error Handling

JSERR (RLU,RLEVEL,ALEVEL)	Set error report conditions
JIERR (ERRNUM,LEVNUM,SUBNUM,INFO)	Inquire last error

JSERR and JIERR enable the application program to control AGP-3's error handling process.

Several types of errors may be generated by an application program. Examples include programming errors, such as attempting to delete a segment that does not exist, and environment errors, such as requesting input from a work station that does not support input. Whenever an error occurs, AGP-3 creates and logs an error message. A complete listing of possible errors is contained in Appendix A.

All errors are assigned a severity level from 0 to 4, where level 0 denotes a warning and level 4 denotes a fatal error. When the error is non-fatal, AGP-3 will attempt recovery before returning to the calling program. The recovery steps taken for specific errors are documented in Appendix A.

JSERR specifies the level of error that is to be reported and the level of error that causes the graphics system to stop processing. An error must be of severity greater than or equal to the error report level (RLEVEL) or it will not be reported. Initially, the error report level is 0, thus, all errors will be logged to RLU, the error report LU. If the error report level is set to 4, the highest severity level, only level 4 errors are reported. RLU is initially set to the LU associated with the scheduling program.

ALEVEL specifies the error abort level. Errors of a severity level greater than or equal to ALEVEL will cause the graphics system to stop processing. When ALEVEL is 0, all errors and warnings are fatal; when ALEVEL is 4, the recoverable errors (levels 0-3) will not cause discontinuation of processing. At initialization, ALEVEL is 4. Once the graphics system has discontinued processing due to an error equal to or exceeding ALEVEL, all AGP-3 calls will be ignored except for JIERR, JBEGN, and JEND.

JIERR returns the most recent error detected by AGP-3. ERRNUM is the error number and LEVNUM denotes the severity level of the last error reported. The number of the routine that detected the error is given by SUBNUM. Although INFO generally provides additional information about the error, for some errors INFO will be undefined and no additional information will be provided. See Appendix A for descriptions of INFO.

As a general programming practice, it is not a good idea to call JIERR after every AGP-3 function. The mechanism by which this information is returned forces an implicit flush of the buffers used in system buffering. This may have a negative effect on system and application performance.

Once AGP-3 has discontinued processing, it retains the error information that caused it to do so. JIERR can be called to determine that error. This call to JIERR must be made correctly because AGP-3 will not attempt to process it if it is made incorrectly.

Appendix A
Error Messages

Introduction

This appendix provides both a table of all AGP-3 subroutine numbers and a listing of the AGP-3 errors. An error message written to the error log LU always includes the number of the error, the number of the subroutine that detected the error, the severity of the error and the additional information provided with each error (if it is defined). This appendix should help the user pinpoint the probable cause of the error based on the subroutine name and error number.

Error Message Format

1) Error Number

2) Error fatality level which ranges from 0 to 4, with 4 being the most severe. The error levels are generally defined as:

   Level 0: Warning. No damage occurred. Examples: No JBEGN; A work station terminated while it was still enabled.

   Level 1: Error. Call ignored but subsequent calls should not be affected. Examples: Attempting to terminate a work station that was never initialized or attempting to turn on the highlighting of an unknown segment.

   Level 2: Error. AGP-3 is able to use defaults, or make implicit calls and then continue the call. Subsequent calls may be affected. Examples: Marker is out of range, default is used; JBATC was called while immediate visibility was in effect. An implicit call is made to JIVOF.

   Level 3: Error. Call ignored. Unexpected results may occur, or some future action may prove impossible. Examples: Attempting to do input from a work station that has never been initialized or attempting to open a segment with an illegal segment name.

Note: Any implicit calls given prior to this error (calls to correct level 2 errors) will be completed even though this call is ignored. For example: Attempting to open a segment that is already open will result in the first segment being closed (error #9 level 2) and the next not being opened due to the duplicate name (error #32 level 3).

Level 4: Disaster. AGP-3 terminates graphics processing. Example: Someone aborted the AGP-3 work station program.

3) Error Message

4) Action taken

5) Explanation of the return parameter INFO. This parameter is printed as part of the error message when the parameter is defined. The application program can access it through JIERR. It provides additional information applicable to a particular error.

6) List of subroutines which report the error.

Using the default severity level of 4, the following applies. If the error severity level is equal to 4, subsequent graphics calls will not be processed. After a level 0 or level 2 error, subsequent errors are tested. After a level 1 or level 3 error, control immediately returns to the caller so subsequent errors are not tested.

If an error generates an implicit call to an AGP-3 routine, errors may be generated in that routine just as if the application program had explicitly made the call. For example: The user makes a call to JWOFF while inside a segment. This is an error and results in an implicit call to JCLOS. Errors could possibly be generated in the JCLOS call.

AGP-3 SUBROUTINE NUMBERS

1 - J2DRW
2 - J2MOV
3 - J2MRK
4 - J2PLY
5 - J3DRW
6 - J3MOV
7 - J3MRK
8 - J3PLY
9 - JASPK
10 - JBATC
11 - JBEGN
12 - JBUTN
13 - JCLOS
14 - JCLPD
15 - JCLPW
16 - JCLR
17 - JCMOD
18 - JCORI
19 - JDDEV
20 - JDFNT
21 - JDINT
22 - JDMOD
23 - unused
24 - JDPTH
25 - JCSIZ
26 - JEDEV
27 - JEND
28 - JALPH
29 - JGDET
30 - JGHI
31 - JGVIS
32 - JHAND
33 - JI1IN
34 - JI1RE
35 - JI2RE
36 - JI3RE
37 - JI4RE
38 - JICP
39 - JIERR
40 - JIESC
41 - JIMAT
42 - unused
43 - JISGA
44 - JISGW
45 - unused
46 - JITSZ
47 - JIVOF
48 - JIVON
49 - JIWND
50 - JIWS
51 - JJUST
52 - JKYBD
53 - unused
54 - JMCUR
55 - unused
56 - JMINT
57 - JNEWF
58 - JOESC
59 - JOPEN
60 - unused
61 - JPICK
62 - JPKID
63 - JPROJ
64 - JPURG
65 - JR2DR
66 - JR2MK
67 - JR2MV
68 - JR2PL
69 - JR3DR
70 - JR3MK
71 - JR3MV
72 - JR3PL
73 - unused
74 - JRNAM
75 - JRSET
76 - JSDET
77 - JSDF
78 - JSERR
79 - JSHI
80 - JSLOC
81 - JSVAL
82 - JSVIS
83 - JTEXH
84 - JTEXL
85 - JTEXM
86 - JUPDT
87 - JVDIS
88 - JVIEW
89 - JVPLN
90 - JVREF
91 - JVSAL
92 - JVTOW
93 - JWEND
94 - JWIND
95 - JWLOC
96 - JWOFF
97 - JWON
98 - JWTOV
99 - JWVAL
100 - JLWID
101 - JLSTL
102 - JCOLR
103 - JDLIM
104 - JLLIM
105 - JPLIM
106 - JLOCP
107 - JPIKP
108 - JDPMM
109 - JLPMM
110 - JPPMM
111 - unused
112 - JFONT

Error Messages

-1XXX Level 3: File manager error number -XXX. The file manager error is due to a problem with the segment display area extension.

INFO: ID of the work station where the error occurred.
ACTION: The work station where the error occurred is terminated. If no work stations are left enabled, then the open segment (if there is one) is closed, and batch of updates (if it is in effect) is terminated.

-2XXX Level 3: File manager error number -XXX. The file manager error is due to a problem with the font file.

INFO: Undefined.
ACTION: Call ignored.

-3XXX Level 3: File manager error -XXX. The file manager error occurred while trying to access the work station program specified.

INFO: ID of the specified work station.
ACTION: Call ignored.

-4XXX Level 3: File manager error number -XXX. The file manager error is due to a problem with the segment display area.

INFO: ID of the work station.
ACTION: Call ignored.

-5XXX Level 3: File manager error number -XXX. The file manager error is due to a problem in opening the intermediate vector file associated with a graphics output device.

INFO: ID of the specific work station.
ACTION: Call ignored.

-6XXX Level 3: File manager error number -XXX. The file manager error is due to a problem with the intermediate vector file associated with a graphics display device.

INFO: ID of the work station where the error occurred.

ACTION: The work station where the error occurred is terminated. If no work stations are left enabled, then the open segment (if there is one) is closed, and batch of updates (if it is in effect) is terminated.

001 Level 3: A non-existent segment was referenced. This segment name was either never created, purged, or renamed.

INFO: Name of specified segment.
ACTION: Call ignored.
REPORTED: JSDET, JRNAM

002 Level 1: A non-existent segment was referenced. This segment name was either never created, purged or renamed.

INFO: Name of specified segment.
ACTION: Call ignored.
REPORTED: JSHI, JPURG, JSVIS

003 Level 3: This routine should not be called during a batch of updates. Possibly missing a JUPDT.

INFO: Undefined.
ACTION: An implicit call is made to JUPDT.
REPORTED: JBATC, JBUTN, JIVON, JKYBD, JMCUR, JPICK, JSLOC, JSVAL, JWLOC, JDINT, JWEND, JWVAL, JIESC, JOESC, JWON, JWOFF, JDDEV, JEDEV, JALPH

004 Level 2: Marker number out of range. Marker numbers can only be in the range 1 to 255. Possibly a negative or real number was passed.

INFO: Marker number requested
ACTION: Marker number 1 is used.
REPORTED: J2MRK, J3MRK, JR2MK, JR3MK

005 Level 3: The work station specified has a dummy device handler loaded for that logical device. An inquiry (see JIWS) of the work station may be made to determine its capabilities.

INFO: ID of the work station.
ACTION: Call ignored for that work station.
REPORTED: JEDEV

006 Level 1: The character plane vector was specified as (0.0,0.0,0.0). Since the vector is specified relative to the origin it is impossible to estabish a vector.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JCORI

007 LEVEL 2: This routine should not be called during immediate visibility. A call to JIVOF may be missing.

INFO: Undefined.
ACTION: An implicit call is made to JIVOF.
REPORTED: JBATC, JIVON

008 Level 1: A character width/height less than or equal to zero was specified. The width and the height must be positive real numbers. Possibly integer values were passed.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JCSIZ

009 Level 2: This routine should not be called while a segment is open. Possibly a call to JCLOS was omitted.

INFO: Name of the open segment.
ACTION: An implicit call is made to JCLOS.
REPORTED: JCLPD, JCLPW, JCMOD, JDINT, JDMOD, JDPTH, JPROJ, JRSET, JVDIS, JVIEW, JVPLN, JVREF, JWEND, JWIND, JWOFF, JWON, JIESC, JOESC, JOPEN, JBATC, JDLIM, JUPDT

010 Level 1: Attribute out of range. Possibly a negative or a real number was passed.

INFO: Attribute value requested.
ACTION: Call ignored.
REPORTED: JLSTL, JLWID, JCOLR

011 Level 3: Unable to compute the composite viewing transformation matrix. Probably the range or the size of some of the viewing parameters are too large or too small for the computer to do the necessary calculations. This error might also occur if AGP-3 code was overwritten.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JOPEN, J2DRW, J2MOV, J2MRK, J2PLY, J3DRW, J3MOV, J3MRK, J3PLY, JR2DR, JR2MV, JR2MK, JR2PL, JR3DR, JR3MV, JR3MK, JR3PL, JTEXH, JTEXL, JTEXM, JICP, JIMAT, JIWND, JVTOW, JWTOV

012 Level 3: The logical class is out of range.

INFO: Logical class requested.
ACTION: Call ignored.
REPORTED: JEDEV

013 Level 2: The open segment has overflowed on the specified work station.

INFO: ID of the work station where overflow occured.
ACTION: Call ignored on that work station.
REPORTED: JSVIS, JSHI, JSDET, JRNAM

014 Level 3: FONT number out of range. Possibly a negative or real number was passed.

INFO: FONT number requested.
ACTION: Call ignored.
REPORTED: JFONT

015 Level 1: The logical class is out of range.

INFO: Logical class requested.
ACTION: Call ignored.
REPORTED: JDDEV

016 Level 3: The pick-ID number is out of range. Possibly a negative or real number was passed.

INFO: Pick-ID number requested.
ACTION: Call ignored.
REPORTED: JPKID

017 Level 3: The logical function requested has not been enabled. Possibly missing a call to JEDEV.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JBUTN, JKYBD, JPICK, JSLOC, JSVAL, JWVAL, JWLOC

018 Level 1: The specified logical function has not been enabled. Possibly missing a call to JEDEV.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDDEV, JALPH, JLLIM, JPLIM, JLOCP, JPIKP, JLPMM, JPPMM, JDPMM

019 Level 2: The open segment has overflowed on the specified work station.

INFO: ID of the work station where the overflow occurred.
ACTION: Call ignored for the open segment on that work station.
REPORTED: JSVAL

020 Level 0: This routine should not be called during immediate visibility. Possibly missing a call to JIVOF.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JMCUR

021 Level 0: A segment was left open. Possibly missing a call to JCLOS.

INFO: Name of the open segment.
ACTION: An implicit call is made to JCLOS.
REPORTED: JEND

022 Level 0: A batch of updates was left open. Possibly missing a call to JUPDT.

INFO: Undefined.
ACTION: An implicit call is made to JUPDT.
REPORTED: JEND

023 Level 1: ID out of range.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JIWS, JLOCP, JALPH, JOESC, JWEND, JWOFF, JDLIM, JDPMM, JIESC, JLLIM, JLPMM, JPIKP, JPLIM, JPPMM

024 Level 2: AGP-3 has already been initialized. Possibly a call to JEND is missing.

INFO: Undefined.
ACTION: An implicit call is made to JEND before reinitializing.
REPORTED: JBEGN

025 Level 3: The maximum number of characters was specified greater than the system maximum.

INFO: The size requested.
ACTION: Call ignored.
REPORTED: JKYBD

026 Level 3: The SDF size given is less than zero. Only non-negative integer numbers should be passed.

INFO: The size given.
ACTION: Call ignored.
REPORTED: JSDF

027 Level 3: The maximum is less than or equal to zero. Only positive integer numbers should be passed.

INFO: The maximum requested.
ACTION: Call ignored.
REPORTED: JKYBD

028 Level 1: ISIZE and/or RSIZE is greater than the number of integer or real parameters required.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JOESC

029 Level 1: This routine should only be called while a segment is open. Possibly a call to JOPEN was omitted or another AGP-3 call did an implicit JCLOS due to an error condition.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JCLOS

030 Level 3: The viewplane normal vector was specified as (0.0,0.0,0.0). Therefore, it is not possible to establish the viewplane. Possibly integer values were passed.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JVPLN

031 Level 1: A segment display area overflow occured during the batch of updates. Possibly making a call to JSDF or purging segments will increase the amount of available storage.

INFO: ID of the work station where overflow occured.
ACTION: The output to be "batched" was sent to the graphical output device before the JUPDT call was executed.
REPORTED: JUPDT

032 Level 3: Attempt to use a segment name that already exists. A segment name must not be bound to any work stations (including work stations that are disabled for graphical output) if it is going to be used for a new segment.

INFO: Name of specified segment.
ACTION: Call ignored.
REPORTED: JOPEN, JRNAM

033 Level 3: An illegal segment name was used. The name must be an integer value in the range 1 to 32767.

INFO: The name requested.
ACTION: Call ignored.
REPORTED: JISGW, JOPEN, JRNAM, JISGA, JPURG, JSDET, JSVIS, JSHI

034 Level 1: The components of the character base vector were specified as (0.0.,0.0,0.0). Since the character base vector is specified relative to the origin, it is impossible to establish a vector.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JCORI

035 Level 3: An enabled work station does not support segment or batch-of-update operations.

INFO: The ID of the work station which does not support the needed operation.
ACTION: Call ignored.
REPORTED: JOPEN, JSDF

036 Level 3: Unable to create intermediate vector file associated with a graphics display device. All available file names are already used. See the GRAPHICS/1000-II Device Handlers Manual for additional details.

INFO: ID of the specific work station.
ACTION: Call ignored.
REPORTED: JDINT

037 Level 4: NTABSZ, in common block &KONTB of the work station program, is not an even, positive number. Possibly the user has incorrectly modified the common block. (NTABSZ defines the number of entries in the hash table for access of segments by name.)

INFO: ID of the specific work station.
ACTION: Call ignored.
REPORTED: JDINT

038 Level X: Unused

039 Level X: Unused

040 Level 1: A polyline routine was called with a negative or zero number of points. Possibly a real number was passed.

INFO: Number of points specified.
ACTION: Call ignored.
REPORTED: J3PLY, J2PLY, JR2PL, JR3PL

041 Level 3: During the construction of the open segment a segment display area overflow occured. Not enough room was left in the SDA to complete the segment. Possibly making a call to JSDF or purging segments will increase the amount of available storage.

INFO: ID of work station that overflowed.
ACTION: The binding between the open segment and the work station is removed. Any primitives on the display (part of the open segment) become primitives outside of a segment. The open segment is closed as normal on all other work stations.
REPORTED: JCLOS

042 Level X: Unused

043 Level 1: The report level is out of range. Possibly a real or negative number was passed.

INFO: Level requested.
ACTION: Call ignored.
REPORTED: JSERR

044 Level 1: The abort level is out of range. Possibly a real or negative number was passed.

INFO: Level requested.
ACTION: Call ignored.
REPORTED: JSERR

045 Level X: Unused

046 Level 1: The character count was less than one, or greater than 132. Possibly a negative or real number was passed.

INFO: Character count specified.
ACTION: Call ignored.
REPORTED: JTEXM, JTEXL, JTEXH, JITSZ, JALPH

047 Level 1: JUPDT was called without batch of updates in effect. Possibly a call to JBATC was omitted, or an error occurred causing the graphics system to close the batch of updates.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JUPDT

048 Level 3: The view up vector was specified as (0.0,0.0,0.0). Therefore, it is not possible to establish the viewplane orientation. Possibly integer values were passed.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JVPLN

049 Level 1: JIVOF was called without immediate visiblity in effect. Possibly a call to JIVON was omitted.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JIVOF

050 Level 3: The viewport boundaries are illegal. The viewport is specified within the bounds of the virtual coordinate system. Therefore, it cannot have values outside the range, (0.0,1.0,0.0,1.0), if the default virtual coordinate system is used. If JASPK was called prior to the JVIEW call, compare the two calls.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JVIEW

051 Level 3: This routine should not be called after the first segment is opened, the first primitive is output, or after a call to JEDEV, JICP, or JIMAT. This call should be moved to the beginning of the program (preferably, right after JBEGN).

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JASPK, JHAND

052 Level X: Unused.

053 Level 3: The viewport boundaries are illegal. Both axes should have a positive non-zero length (i.e., the left value is less than the right value and the bottom value is less than the top value). Possibly integer values were passed.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JVIEW

054 Level 1: ISIZE is not equal to the number of integer parameters to be returned. Possibly a negative or real number was passed.

INFO: ISIZE given.
ACTION: Call ignored.
REPORTED: JIESC, JIWS

055 Level 1: The requested opcode is not available. Possibly a negative or real number was passed.

INFO: Opcode requested.
ACTION: Call ignored.
REPORTED: JI1IN, JI1RE, JI2RE, JI3RE JI4RE, JIESC, JIMAT, JIWS, JOESC

056 Level 3: The window boundaries are illegal. Both axes should have a positive non-zero length (i.e., the left value is less than the right value and the bottom value is less than the top value). Possibly integer values were passed.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JWIND

057 Level X: Unused.

058 Level 2: The center of projection is in or in front of the viewplane. A perspective projection cannot be performed. The center of projection distance (i.e., DN in the call to JPROJ) is calculated relative to the viewplane and should be negative.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JPROJ

059 Level 1: RSIZE is not equal to the number of real parameters to be returned. Possibly a negative or real number was passed.

INFO: RSIZE given.
ACTION: Call ignored.
REPORTED: JIESC, JIWS

060 Level X: Unused.

061 Level 3: The hither plane is behind the yon plane, i.e., HDIST is greater than YDIST.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JDPTH

062 Level 3: The 4-by-4 modelling matrix cannot be inverted.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JDMOD

063 Level 1: ISIZE is not equal to the number of integer parameters required. Possibly a negative or real number was passed.

INFO: ISIZE given.
ACTION: Call ignored.
REPORTED: JOESC

064 Level 0: AGP-3 was not initialized by JBEGN.

INFO: Undefined.
ACTION: An implicit call to JBEGN is made.
REPORTED: Every call but JBEGN and JEND.

065 Level 0: AGP-3 was not initialized by JBEGN.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JEND

066 Level 1: RSIZE is not equal to the number of real parameters required. Possibly a negative or real number was passed.

INFO: RSIZE given.
ACTION: Call ignored.
REPORTED: JOESC

067 Level 1: One or both of the justification values is illegal. They must be between zero and one inclusive.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JJUST

068 Level 0: The work station has not had graphical output disabled. Possibly a call to JWOFF is missing.

INFO: ID of the work station.
ACTION: An implicit call is made to JWOFF.
REPORTED: JWEND

069 Level 0: A work station has not been terminated. Possibly a call to JWEND is missing.

INFO: ID of the work station.
ACTION: An implicit call is made to JWEND.
REPORTED: JEND

070 Level 1: The work station specified has not been initialized. Possibly a call to JDINT is missing, or the ID is not identical to that used in the JDINT call.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDDEV, JIESC, JIWS, JLOCP, JALPH, JOESC, JWEND, JDLIM, JDPMM, JLLIM, JLPMM, JPIKP, JPLIM, JPPMM

071 Level 3: The work station specified has not been initialized by JDINT. Possibly there is a call to JDINT missing, or the ID is not identical to that used in the JDINT call.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JBUTN, JEDEV, JKYBD, JPICK, JSDF, JSLOC, JSVAL, JWLOC, JWON, JWVAL

072 Level 1: The work station specified has already been enabled for graphical output.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JWON

073 Level 0: A logical function of this work station is still enabled. Possibly a call to JDDEV is missing.

INFO: The logical function class number.
ACTION: An implicit call is made to JDDEV.
REPORTED: JWEND

074 Level 3: Work station ID out of range.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JBUTN, JDDEV, JDINT, JEDEV, JKYBD, JPICK, JSDF, JSLOC, JSVAL, JWLOC, JWON, JWVAL

075 Level X: Unused.

076 Level 1: The work station specified by ID has not been enabled for graphical output. Possibly missing a call to JWON.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDLIM, JDPMM, JWOFF, JOESC

077 Level X: Unused.

078 Level 1: No work stations have been enabled for graphics output. This primitive could not be output to any graphics display surface or stored in the segment display area.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: J2DRW, J2MOV, J2PLY, J2MRK, J3DRW, J3MRK, J3PLY, J3MOV, JR2DR, JR2MV, JR2MK, JR2PL, JR3DR, JR3MK, JR3MV, JR3PL, JTEXH, JTEXL, JTEXM

079 Level 3: Class number is not available.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDINT

080 Level 3: Attempt to enable an input function on an outspooled LU. Input can only be performed from a LU other than that of an outspooled graphical display.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JEDEV

081 Level X: Unused.

082 Level 3: Illegal value for OPCODE. The legal value is 0 for a parallel projection and 1 for a perspective projection.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JPROJ

083 Level 3: Illegal value for either HITHR or YON. The legal value is 0 to turn clipping off or 1 to turn clipping on.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JCLPD

084 Level 3: Illegal value for opcode. The legal value is 0 to set the world coordinate system type to "left" and 1 to "right".

INFO: Opcode requested.
ACTION: Call ignored.
REPORTED: JHAND

085 Level X: Unused.

086 Level 3: Illegal value for opcode. The legal value is 0 to turn the function off and 1 to turn the function on.

INFO: Opcode requested.
ACTION: Call ignored.
REPORTED: JCLPW, JCMOD, JGDET, JSDET, JGVIS, JVSAL, JGHI, JSHI, JSVIS

087 Level 1: VX and/or VY is outside of the virtual coordinate viewspace.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JVTOW

088 Level 3: Specified file is not a font file.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JDFNT

089 Level 2: The slant is out of range. The slant must be in the range -PI/2.0 to PI/2.0.

INFO: Undefined.
ACTION: A value of 0.0 is used.
REPORTED: JDFNT

090 Level 3: The font number specified is out of range. Possibly a non-positive number or real number was passed.

INFO: The font requested.
ACTION: Call ignored.
REPORTED: JDFNT

091 Level 3: JSDF can only be called once for each work station. Once the segment display area extension is set up, its size cannot be changed.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JSDF

092 Level 3: The specified work station program cannot be scheduled.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDINT

093 Level 3: The maximum number of work stations for a given CPU have already been initialized. Some other work station must be terminated through a call to JWEND before the specified work station can be initialized.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDINT

094 Level 3: The LU specified does not match the device driver in the WSP, or the device at the LU is down, or the LU is not in the user's session LU table, or the EQT associated with the LU is down. Possibly the wrong driver was loaded upon creation of the WSP or the incorrect LU was specified.

INFO: LU specified.
ACTION: Call ignored.
REPORTED: JDINT, JEDEV

095 Level 2: ECHO parameter out of range. Possibly a negative or real number was passed.

INFO: Echo number requested.
ACTION: Echo 1 is used.
REPORTED: JBUTN, JKYBD, JPICK, JSLOC, JSVAL, JSLOC, JWLOC, JWVAL

096 Level 3: The ID number specified has been used previously for another work station.

INFO: ID number specified.
ACTION: Call ignored.
REPORTED: JDINT

097 Level 0: No work stations have been initialized.

INFO: Undefined.
ACTION: Warning.
REPORTED: JISGA

098 Level X: Unused

099 Level 1: No work stations have been enabled for graphical output.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JOPEN, JBATC

100 Level 4: Class I/O error. Refer to the appropriate operating system manual for a complete description of the error.

INFO: Class I/O error number.
ACTION: Graphics Aborted.

101 Level X: Unused.

102 Level 3: Attempt to enable for spooling a graphics display that does not support spooling.

INFO: Work station ID.
ACTION: Call ignored.
REPORTED: JDINT

103 Level X: Unused.

104 Level X: Unused.

105 Level X: Unused.

106 Level X: Unused.

107 Level X: Unused.

108 Level X: Unused.

109 Level 3: Illegal WSP name.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDINT

110 Level 3: The character base vector and the character plane vectors are parallel. A character plane cannot be established.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JCORI

111 Level 3: The viewplane normal and the view up vector are parallel. A viewplane cannot be established.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JVPLN

112 Level 3: Corrupt font file. One of the x-y character pairs contains illegal information.

INFO: Undefined.
ACTION: Text call aborted.
REPORTED: JTEXH

113 Level 3: All the available cloning names for the WSP are already in use.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDINT

114 Level 3: ID segments are not available to clone the specified WSP.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDINT

115 Level 3: Specified WSP program not found. It has no ID segment and no type 6 file on cartridge 2 or 3. If the program has a type 6 file on a cartridge other than 2 or 3 it should be RP'ed before trying to access it.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDINT

116 Level 4: The monitor program ZMNTR could not be found. ZMNTR must have an ID segment to be scheduled.

INFO: ID of the work station.
ACTION: Graphics aborted.
REPORTED: JDINT, JWEND

117 Level 1: No copy bit set in WSP's ID segment. The specified WSP cannot be cloned. Possibly the WSP program is currently executing.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDINT

118 Level X: Unused

119 Level 3: Checksum error in WSP type 6 file (probably the type 6 program was loaded on another system), or the file is not of type 6.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDINT

120 Level 1: Current font number does not have a font file associated with it. Possibly missing a call to JDFNT or there is no current font i.e., FONT = 0 (see JFONT).

INFO: Current font number
ACTION: Call ignored.
REPORTED: JTEXH, JITSZ

121 Level 2: Attempt to enable a logical device that is already enabled at ID.

INFO: ID of the work station.
ACTION: Previously enabled logical device is disabled with an implicit call to JDDEV.
REPORTED: JEDEV

122 Level 0: There are no work stations currently initialized that are enabled.

INFO: Undefined.
ACTION: Call has no effect (i.e., no work stations are enabled for output).
REPORTED: JWON

123 Level 3: Incorrect number of parameters specified for the call.

INFO: Number of parameters passed.
ACTION: Call ignored.
REPORTED: Every call except JBEGN and JEND.

124 Level 0: No work stations are currently enabled for graphics output.

INFO: Undefined.
ACTION: Work stations were not disabled for graphics output.
REPORTED: JWOFF

125 Level X: Unused.

126 Level 3: Illegal XSIZE and/or YSIZE. They both should be positive real numbers.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JASPK

127 Level X: Unused.

128 Level 3: The view distance is less than zero, implying the view reference point is behind the viewplane.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JVDIS

129 Level X: Unused

130 Level 3: The hither distance and/or the yon distance is less than zero implying the clipping plane is behind the viewplane. The distances must be greater than or equal to zero.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JDPTH

131 Level 1: Dummy graphical output device loaded. Unable to set the echo position on the specified work station.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JLOCP, JPIKP

132 Level 3: An enabled work station does not support batch of update operations because a dummy SDA was loaded.

INFO: ID of the work station that does not support batch of updates.
ACTION: Call ignored.
REPORTED: JBATC

133 Level X: Unused.

134 Level X: Unused.

135 Level 3: Resource numbers are not available to lock the output device.

INFO: The output LU.
ACTION: Call ignored.
REPORTED: JDINT

136 Level 3: The attempt to lock the output LU has failed.

INFO: The output LU.
ACTION: Call ignored.
REPORTED: JDINT

137 Level 3: AGP-3 has detected an abnormal condition such as an abnormal termination of a work station program.

INFO: Undefined.
ACTION: The work station where the error occurred is terminated. If no work stations are left enabled, then the open segment (if there is one) is closed, and batch of udpates (if it is in effect) is terminated.
REPORTED: Can occur at any time.

138 Level 3: AGP-3 has encountered an I/O error, possibly because a WSP has terminated abnormally.

INFO: The number of the I/O error.
ACTION: The work station where the error occurred is terminated. If no work stations are left enabled, then the open segment (if there is one) is closed, and batch of updates (if it is in effect) is terminated.
REPORTED: Can occur at any time.

139 Level 1: The limit boundaries are illegal (i.e., XMIN >= XMAX or YMIN >= YMAX).

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JDLIM, JLLIM, JPLIM

140 Level 1: One or more of the limit boundaries is outside of the physical device limits.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDLIM, JLLIM, JPLIM

141 Level 1: A reference was made to a point outside the current virtual coordinate system.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JLOCP, JPIKP

142 Level 1: Illegal attempt to set the limits of a logical device that is the same physical device as the graphics display.

INFO: Undefined.
ACTION: Call ignored.
REPORTED: JLLIM, JPLIM

143 Level 0: A reference was made to a point outside the current virtual coordinate system.

INFO: Undefined.
ACTION: Data may be invalid.
REPORTED: JWLOC, JSLOC, JPICK, JDPMM, JLPMM, JPPMM

144 Level 3: The work station specified has a dummy device handler loaded for the graphics display device. An inquiry (see JIWS) of the work station may be made to determine its capabilities.

INFO: ID of the work station.
ACTION: Call ignored for that work station.
REPORTED: JWON

145 Level 3: SUBVAL parameter out of range. Possibly a negative or real number was passed.

INFO: SUBVAL requested.
ACTION: Call ignored.
REPORTED: JWVAL, JSVAL

146 Level 4: The monitor program ZMNTR terminated abnormally.

INFO: Undefined.
ACTION: Graphics aborted.
REPORTED: Can occur at any time.

147 Level 3: The WSP encountered a fatal internal error such as a class I/O error.

INFO: Undefined.
ACTION: The work station where the error occurred is terminated. If no work stations are left enabled, then the open segment (if there is one) is closed, and batch of updates (if it is in effect) is terminated.
REPORTED: Can occur at any time.

148 Level 3: The requested work station program is not an AGP-3 work station program.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDINT

149 Level 3: Rev. code mismatch between the UP and the WSP (e.g., the UP was loaded with new software and the WSP was loaded with old software).

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JDINT

150 Level X: Unused.

152 Level X: Unused

153 Level 4: The monitor program ZMNTR terminated abnormally.

INFO: ID of the work station.
ACTION: Graphics aborted.
REPORTED: JDINT, JWEND

154 Level 4: Incorrect ZMNTR for this system. Reload the proper ZMNTR for the operating system.

INFO: Undefined.
ACTION: Graphics aborted.
REPORTED: JDINT, JWEND

155 Level 3: Attempting to do an inquire escape from a work station whose graphical output device is outspooled.

INFO: ID of the work station.
ACTION: Call ignored.
REPORTED: JIESC

156 Level 3: The file size given in the JSDF call is smaller then this work station can use. The file must be at least 4 FMGR blocks larger than the portion of the SDF that is in memory.

INFO: Minimum SDF size that could be used by the work station.
ACTION: Call ignored.
REPORTED: JSDF

157 Level 2: The file size given in the JSDF call is larger than this work station can use.

INFO: Maximum SDF disc file size that could be used by the work station.
ACTION: The SDF disc file size is set to the maximum the work station can use.
REPORTED: JSDF

158 Level 3: An eavesdrop device enabled as the alphanumeric device was terminated because either a data transmission error occurred or the user aborted output.

INFO: ID of the work station at which the error occurred.
ACTION: The work station at which the error occurred is terminated. If no work stations are left enabled, the open segment (if one exists) is closed, and batch of updates (if it is in effect), is terminated.

159 Level 3: An eavesdrop device enabled as the button device was terminated because either a data transmission error occurred or the user aborted output.

INFO: ID of the work station at which the error occurred.

ACTION: The work station at which the error occurred is terminated. If no work stations are left enabled, the open segment (if one exists) is closed, and batch of updates (if it is in effect), is terminated.

160 Level 3: An eavesdrop device enabled as the graphics device was terminated because either a data transmission error occurred or the user aborted output.

INFO: ID of the work station at which the error occurred.

ACTION: The work station at which the error occurred is terminated. If no work stations are left enabled, the open segment (if one exists) is closed, and batch of updates (if it is in effect), is terminated.

161 Level 3: An eavesdrop device enabled as the keyboard device was terminated because either a data transmission error occurred or the user aborted output.

INFO: ID of the work station at which the error occurred.

ACTION: The work station at which the error occurred is terminated. If no work stations are left enabled, the open segment (if one exists) is closed, and batch of updates (if it is in effect), is terminated.

162 Level 3: An eavesdrop device enabled as the locator device was terminated because either a data transmission error occurred or the user aborted output.

INFO: ID of the work station at which the error occurred.

ACTION: The work station at which the error occurred is terminated. If no work stations are left enabled, the open segment (if one exists) is closed, and batch of updates (if it is in effect), is terminated.

163 Level 3: An eavesdrop device enabled as the pick device was terminated because either a data transmission error occurred or the user aborted output.

INFO: ID of the work station at which the error occurred.

ACTION: The work station at which the error occurred is terminated. If no work stations are left enabled, the open segment (if one exists) is closed, and batch of updates (if it is in effect), is terminated.

164 Level 3: An eavesdrop device enabled as the valuator device was terminated because either a data transmission error occurred or the user aborted output.

INFO: ID of the work station at which the error occurred.

ACTION: The work station at which the error occurred is terminated. If no work stations are left enabled, the open segment (if one exists) is closed, and batch of updates (if it is in effect), is terminated.

165-995 Level X: Unused.

996 Level 4: ZMNTR was probably offed at some previous time and is now in an undefined state, or the class number or class buffers that ZMNTR was using were somehow destroyed. The system should be rebooted to return all class numbers, buffers and other resources that ZMNTR was using.

INFO: Used for internal use only
ACTION: Graphics aborted.

997 Level 4: Fatal internal WSP error. Error in the user program may have caused overwriting of AGP-3 code.

INFO: Used for internal use only.
ACTION: Graphics aborted.

998 Level 4: Fatal internal error. Error in the user program may have caused overwriting of AGP-3 code.

INFO: Used for internal use only.
ACTION: Graphics aborted.

Appendix B
Pascal Considerations

The difficulties that arise when accessing AGP-3 routines from a Pascal program are those that arise when accessing any non-Pascal routine from Pascal. This appendix offers suggestions for the Pascal user and solutions for problems that may be encountered. A review of the section of the Pascal/1000 Reference Manual that discusses accessing non-Pascal routines may be appropriate, also.

All AGP-3 routines to be used should be declared as external procedures in the following manner:

```
PROCEDURE AGP_routine
   $ALIAS 'Jxxxx'$
   (VAR param1 : type1;
    VAR param2 : type2;
          :
          :
    VAR paramN : typeN);
EXTERNAL;
```


where "Jxxxx" is the true name of the AGP-3 routine, and "AGP_routine" is a convenient alias. Aliasing is a useful way to allow Pascal's liberal naming conventions to be utilized, and is particularly useful when different data types may be returned by the same routine. Files supplied with the AGP-3 product contain external declarations of these routines.

The actual parameters (param1, param2, ...) passed to AGP-3 routines may be declared VAR, or "call-by-reference". Note that any external routine may modify the actual parameters passed to it. However, AGP-3 modifies only those parameters explicitly documented to have information returned in them. These output parameters should be declared VAR.

The application program must use the data types expected by AGP-3. Unfortunately, Pascal and FORTRAN's data types are often not identical. The responsibility rests with the Pascal user to declare and use appropriate data types for both the formal and actual parameters of AGP-3 routines. One example of this incompatibility is the data type INTEGER. Pascal represents integers in two words, while AGP-3's FORTRAN code stores them in one word. The following declarations match the variable "work_station_id" to the single-word integer that AGP-3 expects:

```
TYPE
     SINGLE_INTEGER  = -32768..32767;

VAR
     work_station_id : SINGLE_INTEGER;
```

Characters must also be represented appropriately. Pascal usually stores one character in a 16-bit word, while AGP-3's FORTRAN places two characters in 16 bits. The application program must, therefore, declare all parameters containing characters to be PACKED ARRAYs of the appropriate size. If an application program were going to pass an array of 40 characters, the variable "text_string" would be declared as follows:

```
TYPE
    CHAR40       = PACKED ARRAY [1..40] OF CHAR;

VAR
    text_string : CHAR40;
```

Some AGP-3 routines pass back information through parameters that could be interpreted as different Pascal data types, depending on the use of the AGP-3 routine. JIWS is an example of such a routine. For opcodes that inquire device information (5050, 5051, 6050, 6051, 7050, 7051 and 8001) the ILIST array parameter to the JIWS routine returns information that for part of the array is to be interpreted as characters, and for the remainder of the array is to be interpreted as integers. If the JIWS routine were to be declared from a Pascal program as it would be from a Fortran program, code would have to be dedicated to translating the character information from the integer array. This would be done in Pascal by either computationally extracting the high and low order bytes of each integer to get the corresponding character or by using a variant record structure that would automatically map the necessary integers to a packed array of characters. An example of this second implementation follows:

```
TYPE
   SINT    =  -32768..32767;        { Single Word Integer }
   LET_NUM = (Alpha, Numeric);

   AGP_STRING = RECORD
          CASE LET_NUM OF
            Alpha   :  (letters : PACKED ARRAY[1..2] OF CHAR);
            Numeric :  (num     : SINT);
        END;  {RECORD}
```

Using this example, each element in the ILIST array would be of type AGP_STRING (i.e. VAR ILIST : ARRAY [1..6] OF AGP_STRING;). If the first element of the array passed back by JIWS contained characters, they could be accessed in the following manner:

```
first_letter := ILIST[1].letters[1];
second_letter := ILIST[1].letters[2];
```

A more straightforward solution to this mapping problem is to define a special AGP-3 routine, aliasing it to JIWS, with the mapping being done transparently. The following type declarations and procedure declaration could be used for the JIWS call with opcodes 5050, 5051, 6050, 6051, 7050 and 7051. (Note that in this example, device_status is the more descriptive variable name for ILIST.)

```
TYPE
   SINT       = -32768 .. 32767;      { Single word integer }
   DEV_STATUS = RECORD
                  name  : PACKED ARRAY [1..6] OF CHAR;
                  state : ARRAY [1..4] OF SINT;
                END;
   REAL_LIST  = ARRAY [1..4] OF REAL;

PROCEDURE inquire_device_status  $ALIAS 'JIWS'$
   (     id,            { id of the work station of the device }
         opcode,        { inquiry opcode                       }
         isize,         { size of the integer array            }
         rsize : SINT;  { size of the real array               }
     VAR device_status : DEV_STATUS;
     VAR dummy         : REAL_LIST  );
```

Unfortunately this more straightforward solution does not work for the JIWS call with an opcode of 8001. In this call the ILIST array elements 4 and 5 (the security code of the extended segment display area file and the disc LU of the extended segment display area file) must be interpreted as a single integer or two characters, depending on the information passed through JSDF to establish these values. The variant record solution presented first is best suited to solve this problem.

It is also possible (as for JSDF, JDFNT and JOESC) that an AGP-3 routine could pass either character or integer information through the same parameter, depending on the desired interpretation of the call. Examples of such a situation exist for file manager level security codes and cartridges since either can be referenced as an integer or a character. In Pascal, an AGP-3 routine could be defined in a variety of ways to gain the desired amount of flexibility. Variant records, as presented in the first solution, could also be used to achieve the desired results.

Table B-1. AGP-3 and Pascal Data Type Correspondence

AGP-3 expects	Pascal data type
Integer	-32768..32767
Character string of n characters	PACKED ARRAY[1..n] OF CHAR
Real	REAL
Array of n integers	ARRAY [1..n] of -32768..32767
Array of n reals	ARRAY [1..n] of real

Table B-1 summarizes the correspondence between AGP-3 parameters and the matching Pascal data type. A file is supplied with the AGP-3 product that contains standard data type declarations for Pascal.

Another consideration to be made when AGP-3 is accessed from Pascal is the type of heap used. The heap is an area from which memory may be allocated at run time; its location relative to the program code is determined by the HEAP compiler option. The heap may exist within the program's logical address space (HEAP 1), in Extended Memory Area (HEAP 2), or may not exist at all (HEAP 0). It is only the second case, HEAP 2, which requires special consideration by AGP-3 users.

When HEAP 2 is used - the heap is in EMA - the addresses of formal parameters are two words long. AGP-3 routines expect their parameters to have single-word addresses, however, the program may force Pascal to use only one word for those parameters' addresses via another compiler option: HEAPPARMS OFF. HEAPPARMS OFF should be set before the declaration of the formal parameters of AGP-3 routines. This tells the compiler to give them single-word addresses. Of course, it also prevents the actual parameters from residing in EMA, which is an unavoidable limitation when non-Pascal routines are used from Pascal.

```
$HEAP 2$

PROGRAM agp3;

TYPE
    SINGLE_INTEGER = -32768..32767;

PROCEDURE get_button
    $ALIAS 'JBUTN'$
    ($HEAPPARMS OFF$
     VAR work_station_id,
         echo_type,
         button : SINGLE_INTEGER
     $HEAPPARMS ON$);
EXTERNAL;
                    :
                    :
```

If either HEAP 0 or HEAP 1 is chosen, HEAPPARMS OFF is unnecessary.

A final difference is that Pascal and FORTRAN do not store array elements in the same order. FORTRAN uses column major order, while Pascal uses row major order. One method of keeping the array indexing correct is to exchange the array subscripts. For example, if the FORTRAN array is

```
INTEGER IA(5,4)
```

then declare the Pascal array as

```
pascal_array [1..4,1..5] OF SINGLE_INTEGER;
```

When AGP-3 wants to receive element IA(2,4) from the Pascal program, it should be passed pascal_array[4,2].

Glossary

Alphanumeric display device
: An output device to which non-graphics text and data may be directed. It is most often used to send prompts, give status and print other messages.

Aspect ratio
: The ratio of the height to width of an area (e.g., the area of a display device, the window, the viewport or the view surface).

Attribute
: See primitive attribute and segment attribute.

Batch of updates
: An AGP-3 timing mode for grouping picture changes so they will not appear on any view surface until a specific termination call is made.

Button input function
: An input function that returns an integer value corresponding to a button or key that is activated.

Center of projection
: The point in the viewing coordinate system to which the projectors from an object converge.

Character base vector
: The vector that determines the direction in which characters advance as they are output.

Character cell
: An imaginary box placed around a character which defines its dimensions. The character size attribute determines the size of the character cell.

Character gap
: The primitive attribute that determines the extra spacing that is added between character cells as a text string is output. Character gap is expressed as a fraction of a character cell width.

Character plane vector
Used in conjunction with the character base vector to specify the orientation of the world coordinate plane in which characters will appear. AGP-3 also uses this vector to determine the character up vector.

Character slant
A primitive attribute which refers to the slant of characters in radians. 0.0 implies no slant, positive values slant characters to the right, and negative values slant characters to the left.

Character up vector
A vector in the character plane defined by AGP-3 to be perpendicular to the character base vector. The character up vector gives the direction of the character cell's sides.

Clipping
The elimination from view of all visible primitives or parts of primitives that lie outside the clipping limits. (See "window clipping" and "depth clipping").

Clipping volume
A three-dimensional subset of the world coordinate system outside of which output primitives will be clipped and not displayed. The clipping volume's sides are determined by the sides of the window, and its ends are the depth clipping planes. If the projection type is parallel, opposite sides of the clipping volume will each be parallel to the other (forming a parallelepiped). If the projection type is perspective, the clipping volume will have the shape of a truncated pyramid, meaning only the top and bottom are parallel to each other.

Current position (CP)
The most recently specified world coordinate point, used as the starting position for graphics output primitives.

Default
See initial value.

Depth clipping
The elimination from view of all primitives or parts of primitives that would appear in front of the hither plane or behind the yon plane so that only the primitives between the two planes appear.

Detectability
A segment attribute that controls the availability of a segment for selection by pick input.

Eavesdrop device
A device connected between the host computer and a terminal which "listens" to the data sent from the host computer. When a special escape sequence is sent, the device accepts all subsequent data and does not pass it on to the terminal. When a second special escape sequence is sent, the device ignores all data and passes it on to the terminal.

Echoing
A mechanism for reflecting the status of an input function. Echoing is manifested in several ways as a function of the different input functions and the different physical devices being used.

Enable a device
Enabling a device makes it available for use. A device must be enabled before requests are directed to it.

Enable for graphics output
Whenever a work station is enabled for graphics output, subsequently created output primitives will be directed to it. When the work station is disabled, no changes will occur on the display device associated with it. Input is not dependent on whether or not a work station is enabled or disabled for graphics output.

Escape function
A facility within AGP-3 that allows access to device-dependent functions.

Fixed-width font
A text font in which every character cell is of the same width.

Font
A primitive attribute applying only to high quality text that determines the style in which characters are drawn. Fonts are stored in a font file.

Font file
A file containing coordinate data and control information which AGP-3 uses to produce high quality text characters.

Gap
See character gap.

GRAPHICS/1000-II Packed ASCII Format
The format in which input characters are returned: ASCII left-justified, one character per byte and blank filled.

Graphics display device
A peripheral which displays graphics output.

High quality text
A level of text quality which adheres exactly to the text attributes requested by the user and is subject to all the viewing operations that apply to line primitives. AGP-3 uses a series of moves and draws to generate high quality text.

Highlighting
A segment attribute used to emphasize the image of a segment. Some devices highlight by blinking or by increasing line intensity, while others ignore the highlighting attribute.

Hither plane
A plane parallel to the viewplane. All primitives or parts of primitives "in front of" the hither plane (i.e., between the hither plane and the center of projection) will be eliminated from view if hither clipping is enabled.

Image
A particular view of one or more objects or parts of objects. Images comprise the picture on a graphics display device.

Immediate visibility
An AGP-3 timing mode in which picture changes requested by an AGP-3 routine are sent to the graphics display device before the routine is exited. This is the least efficient timing mode in system resource usage.

Initial value
The value of an attribute, viewing component, or characteristic of a work station that is in effect when AGP-3 is initialized. In the case of an error, this value is often restored for recovery purposes.

Inquiry
User request for the current status, value, or characteristics of the graphics environment.

Justification
See text justification.

Keyboard input function
An input function that returns a text string of zero or more characters. The typical keyboard device is the alphanumeric keyboard. Keyboard input is terminated when a termination character (e.g. carriage return) is detected.

Left-handed coordinate system
Any three-dimensional coordinate system which can be rotated such that the positive X-axis is directed to the right, the positive Y-axis is directed up, and the positive Z-axis is directed away.

Line
A vector drawn from the current position to a specified point.

Linestyle
An output primitive attribute that controls the pattern with which lines, polylines, and text primitives are drawn.

Linewidth
An output primitive attribute that controls the thickness with which lines, polylines and text primitives are drawn.

Locator input function
An input function that returns a virtual coordinate point corresponding to a location on a locator device.

Logical device
An abstraction of a typical graphics device, defined in terms of the type of data input or output. The logical devices supported by AGP-3 are button, keyboard, locator, valuator, pick, alphanumeric display, and graphics display.

Logical display limits
The bounds of the logical display surface.

Logical display surface
The portion of a graphics display device where all output will appear.

Low quality text
A level of text which uses the device character generator to produce an entire text string as a unit.

Make-picture-current
Action which forces all currently buffered updates to be sent immediately to the display device(s).

Mapping
The transformation of data from one coordinate system to another.

Marker
An output primitive that is a center oriented symbol used to identify a point in the world coordinate system.

Medium quality text
A level of text that uses the device character generator to produce text on a character-by-character basis.

Modelling transformation
The ability to transform an object (e.g., scale or translate it) before the viewing transformation is applied.

Modelling system
A package that provides for generation of complex graphics models of the real world. Modelling implies the manipulation of objects as a whole before a viewing transformation is applied.

Monitor program
A program used to monitor AGP-3 programs and to perform system clean-up (ZMNTR).

Move
Moving the current position to a specified point without generating a line.

New-frame-action
The elimination from view of all primitives outside of segments. On most devices this means the screen will be erased and the visible segments redrawn. On chart advance plotters the paper is advanced and the visible segments are redrawn.

Object
The conceptual graphics entity in the application program. Objects are defined in terms of output primitives and primitive attributes. Their units are the units of the world coordinate system.

Oblique projection
Any projection in which the center of projection is not on the viewplane normal.

Open segment
A segment to which primitives are currently being added. Only one segment may be open at any one time.

Output primitive
The basic element of an object. The output primitives that AGP-3 supports are: move, line, polyline, marker and text. Values of the primitive attributes determine aspects of the appearance of output primitives.

Parallel projection
A viewing projection in which the projectors are parallel. A parallel projection is the special case of a perspective projection in which the center of projection is moved infinitely far from the view reference point.

Perspective projection
A viewing projection in which the center of projection is at a finite distance from the view reference point. The projectors converge to a center of projection in a perspective projection.

Pick-ID
The name given to a primitive or group of primitives within a segment for purposes of pick input.

Pick input function
An input function that returns the name of a visible, detectable segment and a pick-ID within that segment.

Picture
A collective reference to all the images on a display device.

Polyline
An output primitive that defines a connected sequence of lines.

Primitive
See output primitive.

Primitive attribute
A characteristic of an output primitive, such as color, linestyle, text font, etc.

Primitives outside of segments
Output primitives which are not contained in a segment. They do not have segment attributes or a pick-ID associated with them; therefore, they cannot be altered in any way once they have been created. Primitives outside of segments are removed from the graphics display device by a new-frame-action.

Projector
A line which passes from the points on an object through the viewplane to the center of projection. The intersection of the projectors with the viewplane form an object's image.

Requested input
An input operation that requests data and completes when an operator enters a termination command.

Sampled input
An input operation that does not require operator intervention; the routine returns with the current value as soon as the input device can respond.

Segment
A collection of output primitives whose images are treated as a unit. Segments are dynamically modified by altering the segment attributes associated with them. Information on segments is maintained in the segment display area.

Segment attribute
A characteristic of a segment. There are three segment attributes: visibility, highlighting and detectability.

Segment display area (SDA)
A data structure in the work station program which maintains segments. Additionally, during a batch of updates, it is used to maintain primitives outside of segments.

Segment name
An integer name associated with a segment when it is created.

Segment display area extension
An extension to the segment display area created on disc.

Slant
See character slant.

System buffering
An AGP-3 timing mode in which picture changes requested by an AGP-3 routine are subject to buffering.

Text justification
Allows the user to place an entire text string in a desired position relative to the current position.

Text quality
Determines how strictly text attributes will be applied to graphics text. There are three text qualities: high, medium, and low.

Timing mode
Used to control AGP-3 execution efficiency and the immediacy of visual changes to a work station's display device. AGP-3 has three timing modes: immediate visibility, system batching and batch of updates.

User program (UP)
A program written by the user to specify the graphics task. It contains the user written code and the necessary AGP-3 routines.

Valuator input function
An input function that returns a numeric value within the range (0.0 to 1.0) inclusive. The typical valuator is the control dial (potentiometer), but multi-position switches and toggle switches can be utilized as low-precision valuators.

Variable-width font
A text font in which the widths of character cells are unequal.

Viewing coordinate system (UVN system)
The three-dimensional, left-handed coordinate system defined by the view reference point, the viewplane normal, and the viewup vector. The origin is the point where the viewplane normal pierces the viewplane, the positive N-axis is the viewplane normal, the V-axis is the projection (parallel to the viewplane normal) of the view up vector onto the viewplane, and the U-axis is perpendicular to the other two so as to form a left-handed coordinate system.

Viewing operation
See viewing transformation.

Viewing transformation
An operation that maps positions in the world coordinate system to positions in virtual or device coordinates, thereby transforming objects into images.

Viewplane
A plane on which objects are projected to images. In two dimensions, the viewplane is coincident with the X-Y plane.

Viewplane distance
The distance from the view reference point to the viewplane along the viewplane normal. The viewplane distance is always positive.

Viewplane normal vector
A vector specified relative to the view reference point that is perpendicular to the viewplane. The viewplane normal orients the viewplane in the world coordinate system.

Viewport
The rectangular region of the view surface onto which the window will be mapped.

View reference point
A point in the world coordinate system that is used to position the viewing coordinate system.

View surface
The largest rectangle within the logical display limits having the same aspect ratio as the virtual coordinate system.

View up vector
A vector specified relative to the view reference point that, when projected onto the viewplane with a projection parallel to the viewplane normal, defines the "up" direction or V-axis of the viewing coordinate system.

Virtual coordinate system
A two-dimensional coordinate system representing an idealized display device. Virtual coordinates are always in the range 0.0 <= (X,Y) <= 1.0.

Visibility
A segment attribute that indicates whether or not a segment is to have a visible image on the graphics display device.

Window
A rectangular region in the viewplane that may delimit the portion of the projected image that will be output. Projected primitives are clipped to the window's boundaries before they are mapped to the viewport, if window clipping is enabled.

Window clipping
The elimination from view of all primitives or parts of primitives that lie outside the window's boundaries.

Work station
A collection of graphics devices that can be treated as a unit. A work station may have at most one of each available logical device: graphics display, alphanumeric display, button input, keyboard input, locator input, valuator input and pick input.

Work station program (WSP)
A program that receives commands and virtual coordinate data from the user program and performs the graphics tasks specified in the commands. It contains a set of device handlers that perform all graphics I/O to specific devices.

World coordinate system
The three-dimensional Cartesian coordinate system in which objects are described by the user program

Yon plane
A plane parallel to the viewplane. All primitives or parts of primitives "behind" the yon plane (i.e., at a greater distance from the center of projection than the yon plane) will be eliminated from view if yon clipping is enabled.

ZMNTR
See monitor program.

Index

A

B

C

M

N

O

P

V

W

Y

Product Line Sales/Support Key

Key Product Line
A Analytical
CM Components
C Computer Systems
CP Computer Systems Primary Service Responsible Office (SRO)
CS Computer Systems Secondary SRO
E Electronic Instruments & Measurement Systems
M Medical Products
MP Medical Products Primary SRO
MS Medical Products Secondary SRO
P Personal Computing Products
*** Sales only for specific product line**
**** Support only for specific product line**

IMPORTANT: These symbols designate general product line capability. They do not insure sales or support availability for all products within a line, at all locations. Contact your local sales office for information regarding locations where HP support is available for specific products.

HP distributors are printed in italics.

ANGOLA
Telectra
Empresa Técnica de Equipamentos Eléctricos, S.A.R.L.
R. Barbosa Rodrigues, 41-I DT.
Caixa Postal 6487
LUANDA
Tel: 35515,35516
A,E,M,P*

ARGENTINA
Hewlett-Packard Argentina S.A.
Avenida Santa Fe 2035
Martinez 1640 **BUENOS AIRES**
Tel: 798-5735, 792-1293
Telex: 122443 AR CIGY
Cable: HEWPACKARG
A,E,CP,P

Biotron S.A.C.I.y.M
Avenida Paseo Colon 221
9 Piso
1399 ***BUENOS AIRES***
Tel: 30-4846, 30-1851, 30-8384
Telex: (33)17595 BIONAR
Cable: BIOTRON Argentina
M

Fate S.A. Electronica
Bartolomeu Mitre 833
1036 ***BUENOS AIRES***
Tel: 74-41011, 74-49277, 74-43459
Telex: 18137, 22754
P

AUSTRALIA

Adelaide, South Australia Pty. Ltd.
Hewlett-Packard Australia Pty.Ltd.
153 Greenhill Road
PARKSIDE, S.A. 5063
Tel: 272-5911
Telex: 82536
Cable: HEWPARD Adelaide
A*,CM,CS,E,MS,P

Brisbane, Queensland Office
Hewlett-Packard Australia Pty.Ltd.
5th Floor
Teachers Union Building
495-499 Boundary Street
SPRING HILL, Queensland 4000
Tel: 229-1544
Telex: 42133
Cable: HEWPARD Brisbane
A,CM,CS,E,MS,P

Canberra, Australia Capital Office
Hewlett-Packard Australia Pty.Ltd.
121 Wollongong Street
FYSHWICK, A.C.T. 2069
Tel: 804244
Telex: 62650
Cable: HEWPARD Canberra
A*,CM,CS,E,MS,P

Melbourne, Victoria Office
Hewlett-Packard Australia Pty.Ltd.
31-41 Joseph Street
BLACKBURN, Victoria 3130
Tel: 89-6351
Telex: 31-024
Cable: HEWPARD Melbourne
A,CM,CP,E,MS,P

Perth, Western Australia Office
Hewlett-Packard Austalia Pty.Ltd.
141 Stirling Highway
NEDLANDS, W.A. 6009
Tel: 386-5455
Telex: 93859
Cable: HEWPARD Perth
A,CM,CS,E,MS,P

Sydney, New South Wales Office
Hewlett-Packard Australia Pty.Ltd.
17-23 Talavera Road
NORTH RYDE, N.S.W. 2113
P.O. Box 308
Tel: 887-1611
Telex: 21561
Cable: HEWPARD Sydney
A,CM,CP,E,MS,P

AUSTRIA
Hewlett-Packard Ges.m.b.h.
Grottenhofstrasse 94
Verkaufsburo Graz
8052 **GRAZ**
Tel: 21-5-66
Telex: 32375
CM,C*,E*

Hewlett-Packard Ges.m.b.h.
Wehlistrasse 29
P.O. Box 7
A-1205 **VIENNA**
Tel: (222) 35-16-210
Telex: 135823/135066
A,CM,CP,E,MS,P

BAHRAIN
Green Salon
P.O. Box 557
BAHRAIN
Tel: 5503
Telex: 88419
P

Wael Pharmacy
P.O. Box 648
BAHRAIN
Tel: 54886, 56123
Telex: 8550 WAEL GJ
M

BELGIUM
Hewlett-Packard Belgium S.A./N.V.
Blvd de la Woluwe, 100
Woluwedal
B-1200 ***BRUSSELS***
Tel: (02) 762-32-00
Telex: 23-494 paloben bru
A,CM,CP,E,MP,P

BRAZIL
Hewlett-Packard do Brasil I.e.C. Ltda.
Alameda Rio Negro, 750
ALPHAVILLE 06400 Barueri SP
Tel: 421-1311
Telex: 011 23602 HPBR-BR
Cable: HEWPACK Sao Paulo
A,CM,CP,E,MS

Hewlett-Packard do Brasil I.e.C. Ltda.
Avenida Epitacio Pessoa, 4664
22471 **RIO DE JANEIRO**-RJ
Tel: 286-0237
Telex: 021-21905 HPBR-BR
Cable: HEWPACK Rio de Janeiro
A,CM,E,MS,P*

BURUNDI
Typomeca S.P.R.L.
B.P. 553
BUJUMBURA
Tel: 2659
P

CANADA

Alberta
Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher Street S.E.
CALGARY, Alberta T2H 2H8
Tel: (403) 253-2713
Telex: 610-821-6141
A,CM,CP,E*,MS,P*

Hewlett-Packard (Canada) Ltd.
11620A-168th Street
EDMONTON, Alberta T5M 3T9
Tel: (403) 452-3670
Telex: 610-831-2431
A,CM,CP,E,MS,P*

British Columbia
Hewlett-Packard (Canada) Ltd.
10691 Shellbridge Way
RICHMOND, British Columbia V6X 2W7
Tel: (604) 270-2277
Telex: 610-922-5059
A,CM,CP,E*,MS,P*

Manitoba
Hewlett-Packard (Canada) Ltd.
380-550 Century Street
WINNIPEG, Manitoba R3H 0Y1
Tel: (204) 786-6701
A,CM,CS,E,MS,P*

Nova Scotia
Hewlett-Packard (Canada) Ltd.
P.O. Box 931
900 Windmill Road
DARTMOUTH, Nova Scotia B2Y 3Z6
Tel: (902) 469-7820
Telex: 610-271-4482
CM,CP,E*,MS,P*

Ontario
Hewlett-Packard (Canada) Ltd.
552 Newbold Street
LONDON, Ontario N6E 2S5
Tel: (519) 686-9181
Telex: 610-352-1201
A,CM,CS,E*,MS,P*

Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
MISSISSAUGA, Ontario L4V 1M8
Tel: (416) 678-9430
Telex: 610-492-4246
A,CM,CP,E,MP,P

Hewlett-Packard (Canada) Ltd.
1020 Morrison Drive
OTTAWA, Ontario K2H 8K7
Tel: (613) 820-6483
Telex: 610-563-1636
A,CM,CP,E*,MS,P*

Quebec
Hewlett-Packard (Canada) Ltd.
17500 South Service Road
Trans-Canada Highway
KIRKLAND, Quebec H9J 2M5
Tel: (514) 697-4232
Telex: 610-422-3022
A,CM,CP,E,MP,P*

CHILE
Jorge Calcagni y Cia. Ltda.
Arturo Burhle 065
Casilla 16475
SANTIAGO 9
Tel: 220222
Telex: JCALCAGNI
A,CM,E,M

Olympia (Chile) Ltd.
Rodrico de Araya 1045
Casilla 256-V
SANTIAGO *21*
Tel: 25-50-44
Telex: 40-565
C,P

COLOMBIA
Instrumentación
H. A. Langebaek & Kier S.A.
Apartado Aéreo 6287
BOGOTÁ *1, D.E.*
Carrera 7 No. 48-75
BOGOTA*, 2 D.E.*
Tel: 287-8877
Telex: 44400
Cable: AARIS Bogota
A,CM,E,M,P

COSTA RICA
Cientifica Costarricense S.A.
Avenida 2, Calle 5
San Pedro de Montes de Oca
Apartado 10159
SAN JOSÉ
Tel: 24-38-20, 24-08-19
Telex: 2367 GALGUR
Cable: GALGUR
CM,E,M

CYPRUS
Telerexa Ltd.
P.O. Box 4809
14C Stassinos Avenue
NICOSIA
Tel: 45628
Telex: 2894
E,M,P

CZECHOSLOVAKIA
Hewlett-Packard
Obchodni Zastupitelstvi v CSSR
Posl. schranka 27
CS-118 01 ***PRAHA 011***
Tel: 66-296
Telex: 121353 IHC

DENMARK
Hewlett-Packard A/S
Datavej 52
DK-3460 **BIRKEROD**
Tel: (02) 81-66-40
Telex: 37409 hpas dk
A,CM,CP,E,MS,P

Hewlett-Packard A/S
Navervej 1
DK-8600 **SILKEBORG**
Tel: (06) 82-71-66
Telex: 37409 hpas dk
CM,CS,E

ECUADOR
CYEDE Cia. Ltda.
P.O. Box 6423 CCI
Avenida Eloy Alfaro 1749
QUITO
Tel: 450-975, 243-052
Telex: 2548 CYEDE ED
Cable: CYEDE-Quito
A,CM,E,P

Hospitalar S.A.
Casilla 3590
Robles 625
QUITO
Tel: 545-250, 545-122
Cable: HOSPITALAR-Quito
M

EGYPT
Samitro
Sami Amin Trading Office
18 Abdel Aziz Gawish
ABDINE-CAIRO
Tel: 24-932
P

International Engineering Associates
24 Hussein Hegazi Street
Kasr-el-Aini
CAIRO
Tel: 23-829
Telex: 93830
E,M

Informatic For Computer Systems
22 Talaat Harb Street
CAIRO
Tel: 759006
Telex: 93938 FRANK UN
C

EL SALVADOR
IPESA
Boulevard de los Heroes
Edificio Sarah 1148
SAN SALVADOR
Tel: 252787
A,C,CM,E,P

FINLAND
Hewlett-Packard Oy
Revontulentie 7
SF-02100 **ESPOO** 10
Tel: (90) 455-0211
Telex: 121563 hewpa sf
A,CM,CP,E,MS,P

FRANCE
Hewlett-Packard France
Le Ligoures
Bureau de Vente de Aix-en-Provence
Place Romée de Villeneuve
F-13090 **AIX-EN-PROVENCE**
Tel: (42) 59-41-02
Telex: 410770F
A,CM,CS,E,MS,P*

SALES & SUPPORT OFFICES

Arranged alphabetically by country

FRANCE (Cont.)

Hewlett-Packard France
Boite Postale No. 503
F-25026 **BESANCON**
28 Rue de la Republique
F-25000 **BESANCON**
Tel: (81) 83-16-22
C,M

Hewlett-Packard France
Bureau de Vente de Lyon
Chemin des Mouilles
Boite Postale No. 162
F-69130 **ECULLY** Cédex
Tel: (78) 33-81-25
Telex: 310617F
A,CM,CP,E,MP

Hewlett-Packard France
Immeuble France Evry
Tour Lorraine
Boulevard de France
F-91035 **EVRY** Cédex
Tel: (60) 77-96-60
Telex: 692315F
CM,E

Hewlett-Packard France
5th Avenue Raymond Chanas
F-38320 **EYBENS**
Tel: (76) 25-81-41
Telex: 980124 HP GRENOB EYBE
CM,CS

Hewlett-Packard France
Bâtiment Ampère
Rue de la Commune de Paris
Boite Postale 300
F-93153 **LE BLANC MESNIL**
Tel: (01) 865-44-52
Telex: 211032F
CM,CP,E,MS

Hewlett-Packard France
Le Montesquieu
Avenue du President JF Kennedy
F-33700 **MERIGNAC**
Tel: (56) 34-00-84
Telex: 550105F
CM,CP,E,MS

Hewlett-Packard France
32 Rue Lothaire
F-57000 **METZ**
Tel: (87) 65-53-50
CM,CS

Hewlett-Packard France
F-91947 Les Ulis Cédex**ORSAY**
Tel: (1) 907-78-25
Telex: 600048F
A,CM,CP,E,MP,P

Hewlett-Packard France
Paris Porte-Maillot 13, 15 25
Boulevard De L'Amiral Bruix
F-75782 **PARIS** Cédex 16
Tel: (01) 502-12-20
Telex: 613663F
CM,CP,MS,P

Hewlett-Packard France
2 Allee de la Bourgonette
F-35100 **RENNES**
Tel: (99) 51-42-44
Telex: 740912F
CM,CS,E,MS,P*

Hewlett-Packard France
4 Rue Thomas Mann
Boite Postale 56
F-67200 **STRASBOURG**
Tel: (88) 28-56-46
Telex: 890141F
CM,CS,E,MS,P*

Hewlett-Packard France
20 Chemin de la Cépière
F-31081 **TOULOUSE** Cédex
Tel: (61) 40-11-12
Telex: 531639F
A,CM,CS,E,P*

Hewlett-Packard France
Bureau de Vente de Lille
Immeuble Péricentre
Rue Van Gogh
F-59650 **VILLENEUVE D'ASQ**
Tel: (20) 91-41-25
Telex: 160124F
CM,CS,E,MS,P*

GERMAN FEDERAL REPUBLIC

Hewlett-Packard GmbH
Technisches Büro Berlin
Keithstrasse 2-4
D-1000 **BERLIN** 30
Tel: (030) 24-90-86
Telex: 018 3405 hpbln d
A,CM,CS,E,X,M,P

Hewlett-Packard GmbH
Technisches Büro Böblingen
Herrenberger Strasse 110
D-7030 **BÖBLINGEN**
Tel: (07031) 667-1
Telex: 07265739 bbn or 07265743
A,CM,CP,E,MP,P

Hewlett-Packard GmbH
Technisches Büro Dusseldorf
Emanuel-Leutze-Strasse 1
D-4000 **DUSSELDORF**
Tel: (0211) 5971-1
Telex: 085/86 533 hpdd d
A,CM,CP,E,MS,P

Hewlett-Packard GmbH
Vertriebszentrale Frankfurt
Berner Strasse 117
Postfach 560 140
D-6000 **FRANKFURT** 56
Tel: (0611) 50-04-1
Telex: 04 13249 hpffm d
A,CM,CP,E,MP,P

Hewlett-Packard GmbH
Technisches Büro Hamburg
Kapstadtring 5
D-2000 **HAMBURG** 60
Tel: (040) 63804-1
Telex: 21 63 032 hphh d
A,CM,CP,E,MS,P

Hewlett-Packard GmbH
Technisches Büro Hannover
Am Grossmarkt 6
D-3000 **HANNOVER** 91
Tel: (0511) 46-60-01
Telex: 092 3259
A,CM,CS,E,MS,P

Hewlett-Packard GmbH
Technisches Büro Mannheim
Rosslauer Weg 2-4
D-6800 **MANNHEIM**
Tel: (621) 70050
Telex: 0462105
A,C,E

Hewlett-Packard GmbH
Technisches Büro Neu Ulm
Messerschmittstrasse 7
D-7910 **NEU ULM**
Tel:
Telex:
C,E

Hewlett-Packard GmbH
Technisches Büro Nürnberg
Neumeyerstrasse 90
D-8500 **NÜRNBERG**
Tel: (0911) 56-30-83
Telex: 0623 860
CM,CS,E,MS,P

Hewlett-Packard GmbH
Technisches Büro München
Eschenstrasse 5
D-8021 **TAUFKIRCHEN**
Tel: (089) 6117-1
Telex: 0524985
A,CM,CP,E,MS,P

GREAT BRITAIN

Hewlett-Packard Ltd.
Trafalgar House
Navigation Road
ALTRINCHAM
Chesire WA14 1NU
Tel: (061) 928-6422
Telex: 668068
A,C,E,M

Hewlett-Packard Ltd.
Oakfield House, Oakfield Grove
Clifton
BRISTOL BS8 2BN
Tel: 36806
Telex: 444302
P

Hewlett-Packard Ltd.
14 Wesley Street
CASTLEFORD
Yorkshire WF10 1AE
Tel: (0977) 550016
Telex: 5557355
C

Hewlett-Packard Ltd.
Fourier House
257-263 High Street
LONDON COLNEY
Herts., AL2 1HA
Tel: (0727) 24400
Telex: 1-8952716
C,E

Hewlett-Packard Ltd
Tradax House, St. Mary's Walk
MAIDENHEAD
Berkshire, SL6 1ST
Tel: (0628) 39151
E,P

Hewlett-Packard Ltd.
308/314 Kings Road
READING, Berkshire
Tel: 61022
Telex: 84-80-68
CM,P

Hewlett-Packard Ltd.
Quadrangle
106-118 Station Road
REDHILL, Surrey
Tel: (0737) 68655
Telex: 947234 C,E

Hewlett-Packard Ltd.
Westminster House
190 Stratford Road
SHIRLEY, Solihull
West Midlands B90 3BJ
Tel: (021) 7458800
Telex: 339105
C

Hewlett-Packard Ltd.
King Street Lane
WINNERSH, Wokingham
Berkshire RG11 5AR
Tel: (0734) 784774
Telex: 847178
A,C,E,M

GREECE

Kostas Karaynnis
8 Omirou Street
***ATHENS** 133*
Tel: 32-30-303, 32-37-371
Telex: 21 59 62 RKAR GR
E,M,P

"Plaiso"
G. Gerados
24 Stournara Street
ATHENS
Tel: 36-11-160
Telex: 21 9492
P

GUATEMALA

IPESA
Avenida Reforma 3-48
Zona 9
GUATEMALA CITY
Tel: 316627, 314786, 664715
Telex: 4192 Teletro Gu
A,C,CM,E,M,P

HONG KONG

Hewlett-Packard Hong Kong, Ltd.
G.P.O. Box 795
5th Floor, Sun Hung Kai Centre
30 Harbour Road
HONG KONG
Tel: 5-8323211
Telex: 66678 HEWPA HX
Cable: HP ASIA LTD Hong Kong
E,CP,P

Schmidt & Co. (Hong Kong) Ltd.
Wing On Centre, 28th Floor
Connaught Road, C.
HONG KONG
Tel: 5-455644
Telex: 74766 SCHMX HX
A,M

ICELAND

Elding Trading Company Inc.
Hafnarnvoli-Tryggvagotu
P.O. Box 895
IS-REYKJAVIK
Tel: 1-58-20, 1-63-03
M

INDIA

Blue Star Ltd.
Bhavdeep
Stadium Road
***AHMEDABAD** 380 014*
Tel: 42932
Telex: 012-234
Cable: BLUEFROST
E

Blue Star Ltd.
11 Magarath Road
***BANGALORE** 560 025*
Tel: 55668
Telex: 0845-430
Cable: BLUESTAR
A,CM,C,E

Blue Star Ltd.
Band Box House
Prabhadevi
***BOMBAY** 400 025*
Tel: 422-3101
Telex: 011-3751
Cable: BLUESTAR
A,M

Blue Star Ltd.
Sahas
414/2 Vir Savarkar Marg
Prabhadevi
***BOMBAY** 400 025*
Tel: 422-6155
Telex: 011-4093
Cable: FROSTBLUE
A,CM,C,E,M

Blue Star Ltd.
7 Hare Street
***CALCUTTA** 700 001*
Tel: 12-01-31
Telex: 021-7655
Cable: BLUESTAR
A,M

Blue Star Ltd.
Meenakshi Mandiram
XXXXV/1379-2 M. G. Road
***COCHIN** 682-016*
Tel: 32069
Telex: 085-514
Cable: BLUESTAR
*A**

Blue Star Ltd.
133 Kodambakkam High Road
***MADRAS** 600 034*
Tel: 82057
Telex: 041-379
Cable: BLUESTAR
A,M

Blue Star Ltd.
Bhandari House, 7th/8th Floors
91 Nehru Place
***NEW DELHI** 110 024*
Tel: 682547
Telex: 031-2463
Cable: BLUESTAR
A,CM,C,E,M

Blue Star Ltd.
1-1-117/1 Sarojini Devi Road
***SECUNDERABAD** 500 033*
Tel: 70126
Telex: 0155-459
Cable: BLUEFROST
A,E

Blue Star Ltd.
T.C. 7/603 Poornima
Maruthankuzhi
***TRIVANDRUM** 695 013*
Tel: 65799
Telex: 0884-259
Cable: BLUESTAR
E

INDONESIA

BERCA Indonesia P.T.
P.O.Box 496/Jkt.
Jln. Abdul Muis 62
JAKARTA
Tel: 373009
Telex: 31146 BERSAL IA
Cable: BERSAL JAKARTA
A,C,E,M,P

BERCA Indonesia P.T.
P.O. Box 174/Sby.
J.L. Kutei No. 11
SUBAEE-SURABAYA
Tel: 68172
Telex: 31146 BERSAL SD
Cable: BERSAL-SURABAYA
A,E,M,P*

IRAQ

Hewlett-Packard Trading S.A.
Mansoor City 9B/3/7
BAGHDAD
Tel: 551-49-73
Telex: 2455 HEPAIRAQ IK
CP

IRELAND

Hewlett-Packard Ireland Ltd.
Kestrel House
Clanwilliam Court
Lower Mount Street
DUBLIN 2, Eire
Tel: 680424, 680426
Telex: 30439
A,C,CM,E,M,P

Cardiac Services Ltd.
Kilmore Road
Artane
***DUBLIN** 5, Eire*
Tel: (01) 351820
Telex: 30439
M

ISRAEL
Electronics Engineering Division
Motorola Israel Ltd.
16 Kremenetski Street
P.O. Box 25016
TEL-AVIV 67899
Tel: 338973
Telex: 33569 Motil IL
Cable: BASTEL Tel-Aviv
A,CM,C,E,M,P

ITALY
Hewlett-Packard Italiana S.p.A.
Traversa 99C
Giulio Petrone, 19
I-70124 **BARI**
Tel: (080) 41-07-44
M

Hewlett-Packard Italiana S.p.A.
Via Martin Luther King, 38/111
I-40132 **BOLOGNA**
Tel: (051) 402394
Telex: 511630
CM,CS,E,MS

Hewlett-Packard Italiana S.p.A.
Via Principe Nicola 43G/C
I-95126 **CATANIA**
Tel: (095) 37-10-87
Telex: 970291
C,P

Hewlett-Packard Italiana S.p.A.
Via G. Di Vittorio 9
I-20063 **CERNUSCO SUL NAVLIGLIO**
Tel: (2) 903691
Telex: 334632
A,CM,CP,E,MP,P

Hewlett-Packard Italiana S.p.A.
Via Nuova san Rocco A
Capodimonte, 62/A
I-80131 **NAPOLI**
Tel: (081) 7413544
A,CM,CS,E

Hewlett-Packard Italiana S.p.A.
Viale G. Modugno 33
I-16156 **GENOVA PEGLI**
Tel: (010) 68-37-07 E,C

Hewlett-Packard Italiana S.p.A.
Via Turazza 14
I-35100 **PADOVA**
Tel: (49) 664888
Telex: 430315
A,CM,CS,E,MS

Hewlett-Packard Italiana S.p.A.
Viale C. Pavese 340
I-00144 **ROMA**
Tel: (06) 54831
Telex: 610514
A,CM,CS,E,MS,P*

Hewlett-Packard Italiana S.p.A.
Corso Giovanni Lanza 94
I-10133 **TORINO**
Tel: (011) 682245, 659308
Telex: 221079
CM,CS,E

JAPAN
Yokogawa-Hewlett-Packard Ltd.
Inoue Building
1348-3, Asahi-cho
ATSUGI, Kanagawa 243
Tel: (0462) 24-0451
CM,C*,E

Yokogawa-Hewlett-Packard Ltd.
3-30-18 Tsuruya-cho
Kanagawa-ku, Yokohama-Shi
KANAGAWA, 221
Tel: (045) 312-1252
Telex: 382-3204 YHP YOK
CM,CS,E

Yokogawa-Hewlett-Packard Ltd.
Sannomiya-Daiichi Seimei-Bldg. 5F
69 Kyo-Machi Ikuta-Ku
KOBE CITY 650 Japan
Tel: (078) 392-4791
C,E

Yokogawa-Hewlett-Packard Ltd.
Kumagaya Asahi Yasoji Bldg 4F
4-3 Chome Tsukuba
KUMAGAYA, Saitama 360
Tel: (0485) 24-6563
CM,CS,E

Yokogawa-Hewlett-Packard Ltd.
Mito Mitsui Building
4-73, San-no-maru, 1-chome
MITO, Ibaragi 310
Tel: (0292) 25-7470
CM,CS,E

Yokogawa-Hewlett-Packard Ltd.
Sumitomo Seimei Bldg.
11-2 Shimo-sasajima-cho
Nakamura-ku
NAGOYA, Aichi 450
Tel: (052) 581-1850
CM,CS,E,MS

Yokogawa-Hewlett-Packard Ltd.
Chuo Bldg., 4th Floor
5-4-20 Nishinakajima, 5-chome
Yodogawa-ku, Osaka-shi
OSAKA, 532
Tel: (06) 304-6021
Telex: YHPOSA 523-3624
A,CM,CP,E,MP,P*

Yokogawa-Hewlett-Packard Ltd.
29-21 Takaido-Higashi 3-chome
Suginami-ku **TOKYO** 168
Tel: (03) 331-6111
Telex: 232-2024 YHPTOK
A,CM,CP,E,MP,P*

JORDAN
Mouasher Cousins Company
P.O. Box 1387
AMMAN
Tel: 24907, 39907
Telex: 21456 SABCO JO
E,M,P

KOREA
Samsung Electronics
4759 Shinkil, 6 Dong
Youngdeungpo-Ku,
SEOUL
Tel: 8334311, 8334312
Telex: SAMSAN 27364
A,C,E,M,P

KUWAIT
Al-Khalidya Trading & Contracting
P.O. Box 830 Safat
KUWAIT
Tel: 42-4910, 41-1726
Telex: 2481 Areeg kt
A,E,M

Photo & Cine Equipment
P.O. Box 270 Safat
KUWAIT
Tel: 42-2846, 42-3801
Telex: 2247 Matin
P

LUXEMBOURG
Hewlett-Packard Belgium S.A./N.V.
Blvd de la Woluwe, 100
Woluwedal
B-1200 ***BRUSSELS***
Tel: (02) 762-32-00
Telex: 23-494 paloben bru
A,CM,CP,E,MP,P

MALAYSIA
Hewlett-Packard Sales (Malaysia)
Sdn. Bhd.
Suite 2.21/2.22
Bangunan Angkasa Raya
Jalan Ampang
KUALA LUMPUR
Tel: 483544
Telex: MA31011
A,CP,E,M,P*

Protel Engineering
Lot 319, Satok Rd.
P.O. Box 1917
KUCHING, SARAWAK
Tel: 535-44
Telex: MA 70904 Promal
Cable: Proteleng
A,E,M

MEXICO
Hewlett-Packard Mexicana, S.A. de C.V.
Avenida Periferico Sur No. 6501
Tepepan, Xochimilco
MEXICO CITY 23, D.F.
Tel: (905) 676-4600
Telex: 017-74-507
A,CP,E,MS,P

Hewlett-Packard Mexicana, S.A. de C.V.
Rio Volga 600
Colonia del Valle
MONTERREY, N.L.
Tel: 78-42-93, 78-42-40, 78-42-41
Telex: 038-410
CS

MOROCCO
Dolbeau
81 rue Karatchi
CASABLANCA
Tel: 3041-82, 3068-38
Telex: 23051, 22822
E

Gerep
2 rue d'Agadir
Boite Postale 156
CASABLANCA
Tel: 272093, 272095
Telex: 23 739
P

NETHERLANDS
Hewlett-Packard Nederland B.V.
Van Heuven Goedhartlaan 121
NL 1181KK **AMSTELVEEN**
P.O. Box 667
NL1080 AR **AMSTELVEEN**
Tel: (20) 47-20-21
Telex: 13 216
A,CM,CP,E,MP,P

Hewlett-Packard Nederland B.V.
Bongerd 2
NL 2906VK **CAPPELLE**, A/D Ijessel
P.O. Box 41
NL2900 AA **CAPELLE**, Ijssel
Tel: (10) 51-64-44
Telex: 21261 HEPAC NL
A,CM,CP

NEW ZEALAND
Hewlett-Packard (N.Z.) Ltd.
169 Manukau Road
P.O. Box 26-189
Epsom, **AUCKLAND**
Tel: 68-7159
Cable: HEWPACK Auckland
CM,CS,E,P*

Hewlett-Packard (N.Z.) Ltd.
4-12 Cruickshank Street
P.O. Box 9443
Kilbirnie, **WELLINGTON** 3
Tel: 877-199
Cable: HEWPACK Wellington
CM,CP,E,P

Northrop Instruments & Systems Ltd.
Eden House, 44 Khyber Pass Road
P.O. Box 9682
*Newmarket, **AUCKLAND***
Tel: 794-091
A,M

Northrop Instruments & Systems Ltd.
Terrace House, 4 Oxford Terrace
P.O. Box 8388
CHRISTCHURCH
Tel: 64-165
A,M

Northrop Instruments & Systems Ltd.
Sturdee House
85-87 Ghuznee Street
P.O. Box 2406
WELLINGTON
Tel: 850-091
Telex: NZ 3380
A,M

NIGERIA
The Electronics Instrumentations Ltd.
N6B/S70 Oyo Road
Oluseun House
P.M.B. 5402
IBADAN
Tel: 461577
Telex: 31231 TEIL NG
A,E,M,P

The Electronics Instrumentations Ltd.
144 Agege Motor Road, Mushin
P.O. Box 6645
*Mushin, **LAGOS***
A,E,M,P

NORTHERN IRELAND
Cardiac Services Company
95A Finaghy Road South
***BELFAST** BT 10 OBY*
Tel: (0232) 625-566
Telex: 747626
M

NORWAY
Hewlett-Packard Norge A/S
Folke Bernadottesvei 50
P.O. Box 3558
N-5033 **FYLLINGSDALEN** (BERGEN)
Tel: (05) 16-55-40
Telex: 16621 hpnas n
CM,CS,E

Hewlett-Packard Norge A/S
Oesterndalen 18
P.O. Box 34
N-1345 **OESTERAAS**
Tel: (02) 17-11-80
Telex: 16621 hpnas n
A*,CM,CP,E,MS,P

OMAN
Khimjil Ramdas
P.O. Box 19
MUSCAT
Tel: 72-22-17, 72-22-25
Telex: 3289 BROKER MB MUSCAT
P

PAKISTAN
Mushko & Company Ltd.
10, Bazar Road
Sector G-6/4
ISLAMABAD
Tel: 28624
Cable: FEMUS Rawalpindi
A,E,M

Mushko & Company Ltd.
Oosman Chambers
Abdullah Haroon Road
***KARACHI** 0302*
Tel: 511027, 512927
Telex: 2894 MUSHKO PW
Cable: COOPERATOR Karachi
*A,E,M,P**

PANAMA
Electrónico Balboa, S.A.
Aparlado 4929
Panama 5
Calle Samuel Lewis
Edificio "Alfa" No. 2
CIUDAD DE PANAMA
Tel: 64-2700
Telex: 3480380
Cable: ELECTRON Panama
A,CM,E,M,P

Foto Internacional, S.A.
P.O. Box 2068
Free Zone of Colon
COLON 3
Tel: 45-2333
Telex: 3485126
Cable: IMPORT COLON/Panama
P

PERU
Cómpania Electro Médica S.A.
Los Flamencos 145, San Isidro
Casilla 1030
***LIMA** 1*
Tel: 41-4325
Telex: Pub. Booth 25424 SISIDRO
Cable: ELMED Lima
A,CM,E,M,P

PHILIPPINES
The Online Advanced Systems Corporation
Rico House, Amorsolo Cor. Herrera Street
Legaspi Village, Makati
P.O. Box 1510
*Metro **MANILA***
Tel: 85-35-81, 85-34-91, 85-32-21
Telex: 3274 ONLINE
A,C,E,M

Electronic Specialists and Proponents Inc.
690-B Epifanio de los Santos Avenue
*Cubao, **QUEZON CITY***
P.O. Box 2649 Manila
Tel: 98-96-81, 98-96-82, 98-96-83
Telex: 742-40287
P

POLAND
Buro Informasji Technicznej
Hewlett-Packard
Ul Slawki 2, 6P
*PL00-950 **WARSZAWA***
Tel: 39-59-62, 39-67-43
Telex: 812453 hepa pl

SALES & SUPPORT OFFICES
Arranged alphabetically by country

PORTUGAL
Telectra-Empresa Técnica de Equipmentos Eléctricos S.a.r.l.
Rua Rodrigo da Fonseca 103
P.O. Box 2531
***P-LISBON** 1*
Tel: (19) 68-60-72
Telex: 12598
A,C,E,P

Mundinter
Intercambio Mundial de Comércio S.a.r.l
P.O. Box 2761
Avenida Antonio Augusto de Aguiar 138
P-LISBON
Tel: (19) 53-21-31, 53-21-37
Telex: 16691 munter p
M

PUERTO RICO
Hewlett-Packard Puerto Rico
P.O. Box 4407
CAROLINA, Puerto Rico 00630
Calle 272 Edificio 203
Urb. Country Club
RIO PIEDRAS, Puerto Rico 00924
Tel: (809) 762-7255
Telex: 345 0514
A,CP

QATAR
Nasser Trading & Contracting
P.O. Box 1563
DOHA
Tel: 22170
Telex: 4439 NASSER
M

Scitecharabia
P.O. Box 2750
!'DOHA
Tel: 329515
Telex: 4806 CMPARB
P

ROMANIA
Hewlett-Packard Reprezentanta
Boulevard Nicolae Balcescu 16
BUCURESTI
Tel: 130725
Telex: 10440

SAUDI ARABIA
Modern Electronic Establishment
P.O. Box 193
AL-KHOBAR
Tel: 44-678, 44-813
Telex: 670136
Cable: ELECTA AL-KHOBAR
C,E,M,P

Modern Electronic Establishment
P.O. Box 1228, Baghdadiah Street
JEDDAH
Tel: 27-798
Telex: 401035
Cable: ELECTA JEDDAH
C,E,M,P

Modern Electronic Establishment
P.O. Box 2728
RIYADH
Tel: 62-596, 66-232
Telex: 202049
C,E,M,P

SCOTLAND
Hewlett-Packard Ltd.
Royal Bank Buildings
Swan Street
BRECHIN, Angus, Scotland
Tel: 3101, 3102
CM,CS

Hewlett-Packard Ltd.
SOUTH QUEENSFERRY
West Lothian, EH30 9TG
GB-Scotland
Tel: (031) 3311000
Telex: 72682
A,CM,E,M

SINGAPORE
Hewlett-Packard Singapore (Pty.) Ltd.
P.O. Box 58 Alexandra Post Office
SINGAPORE, 9115
6th Floor, Inchcape House
450-452 Alexandra Road
SINGAPORE 0511
Tel: 631788
Telex: HPSGSO RS 34209
Cable: HEWPACK, Singapore
A,CP,E,MS,P

SOUTH AFRICA
Hewlett-Packard South Africa (Pty.) Ltd.
P.O. Box 120
Howard Place
Pine Park Center, Forest Drive, Pinelands
CAPE PROVINCE 7450
Tel: 53-7955, 53-7956, 53-7957
Telex: 57-0006
A,CM,CS,E,MS,P

Hewlett-Packard South Africa (Pty.) Ltd.
P.O. Box 37066
Overport
DURBAN 4067
Tel: 28-4178, 28-4179, 28-4110
CM,CS

Hewlett-Packard South Africa (Pty.) Ltd.
P.O. Box 33345
Glenstantia 0010 **TRANSVAAL**
1st Floor East
Constantia Park Ridge Shopping Centre
Constantia Park
PRETORIA Tel: 98-1126 or 98-1220
Telex: 32163
C,E

Hewlett-Packard South Africa (Pty.) Ltd.
Daphny Street
Private Bag Wendywood
SANDTON 2144
Tel: 802-5111, 802-5125
Telex: 89-84782
Cable: HEWPACK Johannesburg
A,CM,CP,E,MS,P

SPAIN
Hewlett-Packard Española S.A.
c/Entenza, 321
E-BARCELONA 29
Tel: (3) 322-24-51, 321-73-54
Telex: 52603 hpbee
A,CM,CP,E,MS,P

Hewlett-Packard Española S.A.
c/San Vicente S/N
Edificio Albia II,7 B
E-BILBAO 1
Tel: (944) 423-8306, 423-8206
A,CM,E,MS

Hewlett-Packard Española S.A.
Calle Jerez 3
E-MADRID 16
Tel: 458-2600
Telex: 23515 hpe
A,CM,E,MP,P

Hewlett-Packard Española S.A.
Colonia Mirasierra
Edificio Juban
c/o Costa Brava 13, 2.
E-MADRID 34
Tel: 734-8061, 734-1162
CM,CP

Hewlett-Packard Española S.A.
Av Ramón y Cajal 1-9
Edificio Sevilla 1,
E-SEVILLA 5
Tel: 64-44-54, 64-44-58
Telex: 72933
A,CM,CS,MS,P

Hewlett-Packard Española S.A.
C/Ramon Gordillo, 1 (Entlo.3)
E-VALENCIA 10
Tel: 361-1354, 361-1358
CM,CS,P

SWEDEN
Hewlett-Packard Sverige AB
Enighetsvägen 3, Fack
P.O. Box 20502
S-16120 **BROMMA**
Tel: (08) 730-0550
Telex: (854) 10721 MESSAGES
Cable: MEASUREMENTS STOCKHOLM
A,CM,CP,E,MS,P

Hewlett-Packard Sverige AB
Sunnanvagen 14K
S-22226 **LUND**
Tel: (46) 13-69-79
Telex: (854) 10721 (via BROMMA office)
CM,CS

Hewlett-Packard Sverige AB
Vastra Vintergatan 9
S-70344 **OREBRO**
Tel: (19) 10-48-80
Telex: (854) 10721 (via BROMMA office)
CM,CS

Hewlett-Packard Sverige AB
Frötallisgatan 30
S-42132 **VÄSTRA-FRÖLUNDA**
Tel: (031) 49-09-50
Telex: (854) 10721 (via BROMMA office)
CM,CS,E,P

SWITZERLAND
Hewlett-Packard (Schweiz) AG
Clarastrasse 12
CH-4058 **BASLE**
Tel: (61) 33-59-20
A,CM

Hewlett-Packard (Schweiz) AG
47 Avenue Blanc
CH-1202 **GENEVA**
Tel: (022) 32-30-05, 32-48-00
CM,CP

Hewlett-Packard (Schweiz) AG
29 Chemin Château Bloc
CH-1219 **LE LIGNON**-Geneva
Tel: (022) 96-03-22
Telex: 27333 hpag ch
Cable: HEWPACKAG Geneva
A,CM,E,MS,P

Hewlett-Packard (Schweiz) AG
Zürcherstrasse 20
Allmend 2
CH-8967 **WIDEN**
Tel: (57) 50-111
Telex: 59933 hpag ch
Cable: HPAG CH
A,CM,CP,E,MS,P

SYRIA
General Electronic Inc.
Nuri Basha-Ahnat Ebn Kays Street
P.O. Box 5781
DAMASCUS
Tel: 33-24-87
Telex: 11215 ITIKAL
Cable: ELECTROBOR DAMASCUS
E

Sawah & Co.
Place Azmé
Boite Postale 2308
DAMASCUS
Tel: 16-367, 19-697, 14-268
Telex: 11304 SATACO SY
Cable: SAWAH, DAMASCUS
M

TAIWAN
Hewlett-Packard Far East Ltd.
Kaohsiung Branch
68-2, Chung Cheng 3rd Road
Shin Shin, Chu
KAOHSIUNG
Tel: 24-2318, 26-3253
CS,E,MS,P

Hewlett-Packard Far East Ltd.
Taiwan Branch
5th Floor
205 Tun Hwa North Road
TAIPEI
Tel:(02) 751-0404
Cable:HEWPACK Taipei
A,CP,E,MS,P

Hewlett-Packard Far East Ltd.
Taichung Branch
#33, Cheng Yih Street
10th Floor, Room 5
TAICHUNG
Tel: 289274

Ing Lih Trading Co.
3rd Floor 18, Po-la Road
TAIPEI
Tel:
Telex:
Cable: INGLIH TAIPEI
A

THAILAND
UNIMESA Co. Ltd.
Elcom Research Building
2538 Sukhumvit Ave.
*Bangchak, **BANGKOK***
Tel: 393-2387, 393-0338
Telex: TH81160, 82938, 81038
Cable: UNIMESA Bangkok
A,C,E,M

Bangkok Business Equipment Ltd.
5/5-6 Dejo Road
BANGKOK
Tel: 234-8670, 234-8671, 234-8672
Cable: BUSIOUIPT Bangkok
P

TRINIDAD & TOBAGO
Caribbean Telecoms Ltd.
P.O. Box 732
50/A Jerningham Avenue
PORT-OF-SPAIN
Tel: 624-4213, 624-4214
A,CM,E,M,P

TUNISIA
Tunisie Electronique
31 Avenue de la Liberte
TUNIS
Tel: 280-144
E,P

Corema
1 ter. Av. de Carthage
TUNIS
Tel: 253-821
Telex: 12319 CABAM TN
M

TURKEY
Teknim Company Ltd.
Riza Sah Pehlevi
Caddesi No. 7
*Kavaklidere, **ANKARA***
Tel: 275800
Telex: 42155
E

EMA, Muhendislik Kollektif Sirketi
Mediha Eldem
Sokak 41/6
*Yüksel Caddesi, **ANKARA***
Tel: 17-56-22
Cable: Ematrade
M

UNITED ARAB EMIRATES
Emitac Ltd.
P.O. Box 1641
SHARJAH
Tel: 354121, 354123
Telex: 68136
E,M,P,C

UNITED KINGDOM
see: GREAT BRITAIN
NORTHERN IRELAND
SCOTLAND

UNITED STATES
Alabama
Hewlett-Packard Co.
700 Century Park South
Suite 128
BIRMINGHAM, AL 35226
Tel: (205) 822-6802
CM,CS,MP

Hewlett-Packard Co.
P.O. Box 4207
8290 Whitesburg Drive, S.E.
HUNTSVILLE, AL 35802
Tel: (205) 881-4591
CM,CP,E,M*

Alaska
Hewlett-Packard Co.
1577 "C" Street, Suite 252
ANCHORAGE, AK 99510
Tel: (206) 454-3971
CM,CS**

Arizona
Hewlett-Packard Co.
2336 East Magnolia Street
PHOENIX, AZ 85034
Tel: (602) 273-8000
A,CM,CP,E,MS

Hewlett-Packard Co.
2424 East Aragon Road
TUCSON, AZ 85702
Tel: (602) 889-4631
CM,CS,E,MS**

Arkansas
Hewlett-Packard Co.
P.O. Box 5646
Brady Station
LITTLE ROCK, AR 72215
Tel: (501) 376-1844, (501) 664-8773
CM,MS

UNITED STATES (Cont.)

California
Hewlett-Packard Co.
7621 Canoga Avenue
CANOGA PARK, CA 91304
Tel: (213) 702-8300
A,CM,CP,E,P

Hewlett-Packard Co.
1579 W. Shaw Avenue
FRESNO, CA 93771
Tel: (209) 224-0582
CM,MS

Hewlett-Packard Co.
1430 East Orangethorpe
FULLERTON, CA 92631
Tel: (714) 870-1000
CM,CP,E,MP

Hewlett-Packard Co.
5400 W. Rosecrans Boulevard
LAWNDALE, CA 90260
P.O. Box 92105
LOS ANGELES, CA 90009
Tel: (213) 970-7500
CM,CP,MP

Hewlett-Packard Co.
3939 Lankershim Blvd.
NORTH HOLLYWOOD, CA 91604
Tel: (213) 877-1282
Regional Headquarters

Hewlett-Packard Co.
3200 Hillview Avenue
PALO ALTO, CA 94304
Tel: (415) 857-8000
CM,CP,E

Hewlett-Packard Co.
646 W. North Market Boulevard
SACRAMENTO, CA 95834
Tel: (916) 929-7222
A*,CM,CP,E,MS

Hewlett-Packard Co.
9606 Aero Drive
P.O. Box 23333
SAN DIEGO, CA 92123
Tel: (714) 279-3200
CM,CP,E,MP

Hewlett-Packard Co.
3003 Scott Boulevard
SANTA CLARA, CA 95050
Tel: (408) 988-7000
A,CM,CP,E,MP

Hewlett-Packard Co.
454 Carlton Court
SO. SAN FRANCISCO, CA 94080
Tel: (415) 877-0772
CM,CP

Colorado
Hewlett-Packard Co.
24 Inverness Place, East
ENGLEWOOD, CO 80112
Tel: (303) 771-3455
A,CM,CP,E,MS

Connecticut
Hewlett-Packard Co.
47 Barnes Industrial Road South
P.O. Box 5007
WALLINGFORD, CT 06492
Tel: (203) 265-7801
A,CM,CP,E,MS

Florida
Hewlett-Packard Co.
P.O. Box 24210
2727 N.W. 62nd Street
FORT LAUDERDALE, FL 33309
Tel: (305) 973-2600
CM,CP,E,MP

Hewlett-Packard Co.
4080 Woodcock Drive, #132
Brownett Building
JACKSONVILLE, FL 32207
Tel: (904) 398-0663
CM,C*,E*,MS**

Hewlett-Packard Co.
P.O. Box 13910
6177 Lake Ellenor Drive
ORLANDO, FL 32809
Tel: (305) 859-2900
A,CM,CP,E,MS

Hewlett-Packard Co.
6425 N. Pensacola Blvd.
Suite 4, Building 1
PENSACOLA, FL 32575
Tel: (904) 476-8422
A,CM,MS

Hewlett-Packard Co.
110 South Hoover, Suite 120
Vanguard Bldg.
TAMPA, FL 33609
Tel: (813) 872-0900
A*,CM,CS,E*,M*

Georgia
Hewlett-Packard Co.
P.O. Box 105005
2000 South Park Place
ATLANTA, GA 30339
Tel: (404) 955-1500
Telex: 810-766-4890
A,CM,CP,E,MP

Hewlett-Packard Co.
Executive Park Suite 306
P.O. Box 816
AUGUSTA, GA 30907
Tel: (404) 736-0592
CM,MS

Hewlett-Packard Co.
P.O. Box 2103
1172 N. Davis Drive
WARNER ROBINS, GA 31098
Tel: (912) 922-0449
CM,E

Hawaii
Hewlett-Packard Co.
Kawaiahao Plaza, Suite 190
567 South King Street
HONOLULU, HI 96813
Tel: (808) 526-1555
A,CM,CS,E,MS

Idaho
Hewlett-Packard Co.
11311 Chinden Boulevard
BOISE, ID 83707
Tel: (208) 376-6000
CM,CS,M*

Illinois
Hewlett-Packard Co.
211 Prospect Road
BLOOMINGTON, IL 61701
Tel: (309) 663-0383
CM,CS,MS**

Hewlett-Packard Co.
1100 31st Street
DOWNERS GROVE, IL 60515
Tel: (312) 960-5760
CM,CP

Hewlett-Packard Co.
5201 Tollview Drive
ROLLING MEADOWS, IL 60008
Tel: (312) 255-9800
A,CM,CP,E,MP

Indiana
Hewlett-Packard Co.
P.O. Box 50807
7301 No. Shadeland Avenue
INDIANAPOLIS, IN 46250
Tel: (317) 842-1000
A,CM,CS,E,MS

Iowa
Hewlett-Packard Co.
2415 Heinz Road
IOWA CITY, IA 52240
Tel: (319) 351-1020
CM,CS,E*,MS

Kansas
Hewlett-Packard Co.
1644 S. Rock
WICHITA, KA 67207
Tel: (316) 265-5200
CM,CS

Kentucky
Hewlett-Packard Co.
10170 Linn Station Road
Suite 525
LOUISVILLE, KY 40223
Tel: (502) 426-0100
A,CM,CS,MS

Louisiana
Hewlett-Packard Co.
P.O. Box 1449
3229 Williams Boulevard
KENNER, LA 70062
Tel: (504) 443-6201
A,CM,CS,E,MS

Maryland
Hewlett-Packard Co.
7121 Standard Drive
HANOVER, MD 21076
Tel: (301) 796-7700
A,CM,CP,E,MS

Hewlett-Packard Co.
2 Choke Cherry Road
ROCKVILLE, MD 20850
Tel: (301) 948-6370
Telex: 710-828-9685
A,CM,CP,E,MP

Massachusetts
Hewlett-Packard Co.
32 Hartwell Avenue
LEXINGTON, MA 02173
Tel: (617) 861-8960
A,CM,CP,E,MP

Michigan
Hewlett-Packard Co.
23855 Research Drive
FARMINGTON HILLS, MI 48024
Tel: (313) 476-6400
A,CM,CP,E,MP

Hewlett-Packard Co.
4326 Cascade Road S.E.
GRAND RAPIDS, MI 49506
Tel: (616) 957-1970
CM,CS,MS

Minnesota
Hewlett-Packard Co.
2025 W. Larpenteur Ave.
ST. PAUL, MN 55113
Tel: (612) 644-1100
A,CM,CP,E,MP

Mississippi
Hewlett-Packard Co.
P.O. Box 5028
322 N. Mart Plaza
JACKSON, MS 39216
Tel: (601) 982-9363
CM,MS

Missouri
Hewlett-Packard Co.
11131 Colorado Avenue
KANSAS CITY, MO 64137
Tel: (816) 763-8000
Telex: 910-771-2087
A,CM,CS,E,MS

Hewlett-Packard Co.
1024 Executive Parkway
ST. LOUIS, MO 63141
Tel: (314) 878-0200
A,CM,CP,E,MP

Nebraska
Hewlett-Packard
7101 Mercy Road
Suite 101, IBX Building
OMAHA, NE 68106
Tel: (402) 392-0948
CM,MS

Nevada
Hewlett-Packard Co.
Suite D-130
5030 Paradise Blvd.
LAS VEGAS, NV 89119
Tel: (702) 736-6610
CM,MS**

New Jersey
Hewlett-Packard Co.
Crystal Brook Professional Building
Route 35
EATONTOWN, NJ 07724
Tel: (201) 542-1384
A*,CM,C*,E*,P*

Hewlett-Packard Co.
W120 Century Road
PARAMUS, NJ 07652
Tel: (201) 265-5000
A,CM,CP,E,MP

Hewlett-Packard Co.
60 New England Avenue West
PISCATAWAY, NJ 08854
Tel: (201) 981-1199
A,CM,CP,E

New Mexico
Hewlett-Packard Co.
P.O. Box 11634
11300 Lomas Blvd.,N.E.
ALBUQUERQUE, NM 87123
Tel: (505) 292-1330
Telex: 910-989-1185
CM,CP,E,MS

New York
Hewlett-Packard Co.
5 Computer Drive South
ALBANY, NY 12205
Tel: (518) 458-1550
Telex: 710-444-4691
A,CM,CS,E,MS

Hewlett-Packard Co.
9600 Main Street
CLARENCE, NY 14031
Tel: (716) 759-8621
Telex: 710-523-1893

Hewlett-Packard Co.
200 Cross Keys Office
FAIRPORT, NY 14450
Tel: (716) 223-9950
Telex: 510-253-0092
CM,CP,E,MS

Hewlett-Packard Co.
No. 1 Pennsylvania Plaza
55th Floor
34th Street & 8th Avenue
NEW YORK, NY 10119
Tel: (212) 971-0800
CM,CP,E*,M*

Hewlett-Packard Co.
5858 East Molloy Road
SYRACUSE NY 13211
Tel: (315) 455-2486
A,CM,CS,E,MS

Hewlett-Packard Co.
3 Crossways Park West
WOODBURY, NY 11797
Tel: (516) 921-0300
Telex: 510-221-2183
A,CM,CP,E,MS

North Carolina
Hewlett-Packard Co.
P.O. Box 15579
2905 Guess Road (27705)
DURHAM, NC 27704
Tel: (919) 471-8466
C,M

Hewlett-Packard Co.
5605 Roanne Way
GREENSBORO, NC 27409
Tel: (919) 852-1800
A,CM,CP,E,MS

Ohio
Hewlett-Packard Co.
9920 Carver Road
CINCINNATI, OH 45242
Tel: (513) 891-9870
CM,CP,MS

Hewlett-Packard Co.
16500 Sprague Road
CLEVELAND, OH 44130
Tel: (216) 243-7300
Telex: 810-423-9430
A,CM,CP,E,MS

Hewlett-Packard Co.
962 Crupper Ave.
COLUMBUS, OH 43229
Tel: (614) 436-1041
CM,CP,E*

Hewlett-Packard Co.
330 Progress Rd.
DAYTON, OH 45449
Tel: (513) 859-8202
A,CM,CP,E*,MS

Oklahoma
Hewlett-Packard Co.
P.O. Box 366
1503 W. Gore Blvd., Suite #2
LAWTON, OK 73502
Tel: (405) 248-4248
C

Hewlett-Packard Co.
P.O. Box 32008
304 N. Meridan Avenue, Suite A
OKLAHOMA CITY, OK 73107
Tel: (405) 946-9499
A*,CM,CP,E*,MS

Hewlett-Packard Co.
Suite 121
9920 E. 42nd Street
TULSA, OK 74145
Tel: (918) 665-3300
A**,CM,CS,M*

Oregon
Hewlett-Packard Co.
1500 Valley River Drive, Suite 330
EUGENE, OR 97401
Tel: (503) 683-8075
C

Hewlett-Packard Co.
9255 S. W. Pioneer Court
WILSONVILLE, OR 97070
Tel: (503) 682-8000
A,CM,CP,E*,MS

Pennsylvania
Hewlett-Packard Co.
1021 8th Avenue
King of Prussia Industrial Park
KING OF PRUSSIA, PA 19406
Tel: (215) 265-7000
Telex: 510-660-2670
A,CM,CP,E,MP

Hewlett-Packard Co.
111 Zeta Drive
PITTSBURGH, PA 15238
Tel: (412) 782-0400
A,CM,CP,E,MP

South Carolina
Hewlett-Packard Co.
P.O. Box 6442
6941-0 N. Trenholm Road
COLUMBIA, SC 29260
Tel: (803) 782-6493
CM,CS,E,MS

SALES & SUPPORT OFFICES
Arranged alphabetically by country

UNITED STATES (Cont.)

South Carolina (Cont.)
Hewlett-Packard Co.
814 Wade Hampton Blvd.
Suite 10
GREENVILLE, SC 29609
Tel: (803) 232-0917
C

Tennessee
Hewlett-Packard Co.
P.O. Box 22490
224 Peters Road
Suite 102
KNOXVILLE, TN 37922
Tel: (615) 691-2371
A*,CM,MS

Hewlett-Packard Co.
3070 Directors Row
MEMPHIS, TN 38131
Tel: (901) 346-8370
A,CM,CS,MS

Hewlett-Packard Co.
Suite 103
478 Craighead Street
NASHVILLE, TN 37204
Tel: (615) 383-9136
CM,MS**

Texas
Hewlett-Packard Co.
Suite 310W
7800 Shoalcreek Blvd.
AUSTIN, TX 78757
Tel: (512) 459-3143
CM,E

Hewlett-Packard Co.
Suite C-110
4171 North Mesa
EL PASO, TX 79902
Tel: (915) 533-3555
CM,CS,E*,MS**

Hewlett-Packard Co.
5020 Mark IV Parkway
FORT WORTH, TX 76106
Tel: (817) 625-6361
CM,C*

Hewlett-Packard Co.
P.O. Box 42816
10535 Harwin Street
HOUSTON, TX 77036
Tel: (713) 776-6400
A,CM,CP,E,MP

Hewlett-Packard Co.
3309 67th Street
Suite 24
LUBBOCK, TX 79413
Tel: (806) 799-4472
M

Hewlett-Packard Co.
P.O. Box 1270
930 E. Campbell Rd.
RICHARDSON, TX 75081
Tel: (214) 231-6101
A,CM,CP,E,MP

Hewlett-Packard Co.
205 Billy Mitchell Road
SAN ANTONIO, TX 78226
Tel: (512) 434-8241
CM,CS,E,MS

Utah
Hewlett-Packard Co.
3530 W. 2100 South Street
SALT LAKE CITY, UT 84119
Tel: (801) 974-1700
A,CM,CP,E,MS

Virginia
Hewlett-Packard Co.
P.O. Box 9669
2914 Hungary Spring Road
RICHMOND, VA 23228
Tel: (804) 285-3431
A,CM,CP,E,MS

Hewlett-Packard Co.
P.O. Box 4786
3110 Peters Creek Road, N.W.
ROANOKE, VA 24015
Tel: (703) 563-2205
CM,CS,E**

Hewlett-Packard Co.
P.O. Box 12778
5700 Thurston Avenue
Suite 111
VIRGINIA BEACH, VA 23455
Tel: (804) 460-2471
CM,CS,MS

Washington
Hewlett-Packard Co.
15815 S.E. 37th Street
BELLEVUE, WA 98006
Tel: (206) 643-4000
A,CM,CP,E,MP

Hewlett-Packard Co.
Suite A
708 North Argonne Road
SPOKANE, WA 99206
Tel: (509) 922-7000
CM,CS

West Virginia
Hewlett-Packard Co.
4604 MacCorkle Ave., S.E.
CHARLESTON, WV 25304
Tel: (304) 925-0492
A,CM,MS

Wisconsin
Hewlett-Packard Co.
150 S. Sunny Slope Road
BROOKFIELD, WI 53005
Tel: (414) 784-8800
A,CM,CS,E*,MP

URUGUAY
Pablo Ferrando S.A.C. e.I.
Avenida Italia 2877
Casilla de Correo 370
MONTEVIDEO
Tel: 403102
Telex: 901 Public Booth Para Pablo Ferrando 919520
Cable: RADIUM Montevideo
A,CM,E,M

Guillermo Kraft del Uruguay S.A.
Avda. Libertador Brig. Gral. Lavalleja 2083
MONTEVIDEO
Tel: 234588, 234808, 208830
Telex: 6245 ACTOUR UY
P

U.S.S.R.
Hewlett-Packard Co.
Representative Office
Pokrovsky Blvd. 4/17 KV12
MOSCOW 101000 Tel: 294-2024
Telex: 7825 HEWPACK SU

VENEZUELA
Hewlett-Packard de Venezuela C.A.
Apartado 50933
3A Transversal Los Ruices Norte
Edificio Segre 2Y3
CARACAS 1071
Tel: 239-4133, 239-4777, 239-4244
Telex: 25146 HEWPACK
Cable: HEWPACK Caracas
A,CP,E,MS,P

YUGOSLAVIA
Iskra-Commerce-Representation of Hewlett-Packard
Sava Centar Delegacija 30
Milentija Popovica 9
11170 ***BEOGRAD***
Tel: 638-762
Telex: 12042, 12322 YU SAV CEN

Iskra-Commerce-Representation of Hewlett-Packard
Koprska 46
61000 ***LJUBLJANA***
Tel: 321674, 315879
Telex:

ZAMBIA
R. J. Tilbury (Zambia) Ltd.
P.O. Box 2792
LUSAKA
Tel: 81243
A,E,M,P

ZIMBABWE
Field Technical Sales
45 Kelvin Road, North
P.B. 3458
SALISBURY
Tel:
C,E,M,P

FOR COUNTRIES AND AREAS NOT LISTED:

CANADA
Ontario
Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
MISSISAUGA, Ontario L4V 1M8
Tel: (416) 678-9430
Telex: 610-492-4246

EASTERN USA
Maryland
Hewlett-Packard Co.
4 Choke Cherry Road
Rockville, MD 20850
Tel: (301) 258-2000

MIDWESTERN USA
Illinois
Hewlett-Packard Co.
5201 Tollview Drive
ROLLING MEADOWS, IL 60008
Tel: (312) 255-9800

SOUTHERN USA
Georgia
Hewlett-Packard Co.
P.O. Box 105005
450 Interstate N. Parkway
ATLANTA, GA 30339
Tel: (404) 955-1500

WESTERN USA
California
Hewlett-Packard Co.
3939 Lankersim Blvd.
LOS ANGELES, CA 91604
Tel: (213) 877-1282

EUROPEAN AREAS NOT LISTED, CONTACT

SWITZERLAND
Hewlett-Packard S.A.
7 Rue du Bois-du-Lan
CH-1217 **MEYRIN 2**, Switzerland
Tel: (022) 83-81-11
Telex: 27835 hpse
Cable: HEWPACKSA Geneve

EAST EUROPEAN AREAS NOT LISTED, CONTACT

AUSTRIA
Hewlett-Packard Ges.m.b.h.
Wehlistrasse 29
P.O. Box 7
A-1205 **VIENNA**
Tel: (222) 35-16-210
Telex: 135823/135066

MEDITERRANEAN AND MIDDLE EAST AREAS NOT LISTED, CONTACT

GREECE
Hewlett-Packard S.A.
Mediterranean & Middle East Operations
35, Kolokotroni Street
Platia Kefallariou
GR-Kifissia, **ATHENS**, Greece
Tel: 808-0359, 808-0429
Telex: 21-6588
Cable: HEWPACKSA Athens

INTERNATIONAL AREAS NOT LISTED, CONTACT

OTHER AREAS
Hewlett-Packard Co.
Intercontinental Headquarters
3495 Deer Creek Road
PALO ALTO, CA 94304
Tel: (415) 857-1501
Telex: 034-8300
Cable: HEWPACK

READER COMMENT SHEET

GRAPHICS/1000-II
AGP-3 User's Guide

92842-90002 **March 1982**

Update No. __________
(If Applicable)

We welcome your evaluation of this manual. Your comments and suggestions help us improve our publications. Please use additional pages if necessary.

FROM:

Name ______________________________

Company ______________________________

Address ______________________________

____

____

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 141 CUPERTINO, CA.

— POSTAGE WILL BE PAID BY —

Hewlett-Packard Company
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014
ATTN: Technical Marketing Dept.

FOLD

FOLD