AVS 5.5 FOR HEWLETT PACKARD

CHAPTER FIVE

Introduction

This chapter describes AVS 5.5 as it runs on Hewlett Packard HP9000/7xx workstations using HP-UX 10.20 (HP1020). It covers hardware and software prerequisites needed to run AVS; differences and special considerations for running AVS on this platform; and known problems.

AVS5.5 NOTE: HP-UX 10.20 now supports OpenGL (1.1) as the default renderer, an addition since AVS 5.4; read this chapter for more information on this new renderer and the features it supports, including stereo. PHIGS is still supported in a secondary *avs.phigs* kernel, which was compiled against a updated version of PHIGS (3.40). The default setting for *avs* is to enable the hardware renderer; previous releases came up with *-nohw* set by default. HP-UX 10.10 is no longer supported; if you require this platform, please request AVS 5.4 from support@avs.com or your local sales office.

NOTE: Additional information on using AVS under X windows, general programming considerations, and bugs fixed in AVS 5.5 can be found in the *AVS 5.5 on UNIX Platforms* chapter above.

Hardware Prerequisites

Workstation Models

AVS will execute on all of the workstations in the HP 9000 7*xx* series.

Graphics Hardware

This release of AVS uses two mechanisms to produce its screen renderings of Geometry Viewer objects and scenes: a **hardware renderer** that uses the native graphics hardware and an interface graphics library (either OpenGL (default) or PHIGS); and a **software renderer** that implements a set of graphics primitives (polygons, lighting model, perspective, shading, color, etc.) in software, rendering the resulting image into an X Window System image.

The default *avs* executable supports the OpenGL hardware renderer. A second *avs.phigs* executable supports PHIGS.

The **software renderer** will work on any supported HP workstation model with an 8- or 24-plane color frame buffer, the XPutImage *xlib* call, and a local X server that supports them. This includes the PVRX graphics adapter.

The **hardware renderer** will work with the CRX, CRX24, CRX24Z, CRX48Z, HCRX-8, HCRX-8Z, HCRX-24, HCRX-24Z, Visualize-8, Visualize-24, Visualize-48, TVRX and Visualize-FX{2,4,6} graphics adapters as well as possibly other built-in graphics adapters that come on HP platforms. See the subsequent section on "Software Prerequisites" for the system software needed to be able to use the hardware rendering capability.

To interactively determine which graphics adapter your HP workstation has, first determine which device file is in use by your primary X display as follows:

```
grep /dev/ /usr/lib/X11/X0screens | grep -v #
```

then type the following command using the device file that is returned, e.g. /dev/crt:

```
graphinfo /dev/crt
```

or

```
graphinfo /dev/crt_hyper
```

The following is a partial listing of sample output from the *graphinfo* command. In this example, the graphics adapter is a VISUALIZE-FX4.

```
HEWLETT PACKARD WORKSTATION GRAPHICS CONFIGURATION

PRODUCT INFORMATION

    graphics product:           HP VISUALIZE-FX4 (Rev. A)
    driver name:                hpvisx
    archive library:            /opt/graphics/common/lib/libddvisx.a
    shared library:             /opt/graphics/common/lib/libddvisx.sl
                                /opt/graphics/OpenGL/lib/libddvisxgl.sl
    device pathname:            /dev/crt
    spu description:            9000/782

CONFIGURATION INFORMATION
```

```
image planes:               48
overlay planes:             8
resolution:                 1280 X 1024
color or grayscale:         color
Powershade installed:       yes
PHIGS supported:            yes
hardware accelerator:       yes
geometry accelerator:       yes
image accelerator:          no
texture accelerator:        yes
hardware zbuffer:           yes
software zbuffer:           no
video out:                  yes
```

Memory

AVS requires that the HP 9000/7*xx* be configured with a minimum of 16 megabytes of main memory. More real memory, such as 32 or 64 megabytes, will improve performance.

To interactively determine the amount of memory that your HP workstation has, log in as *root* and type the command:

```
/etc/dmesg
```

This command generates a listing of various system messages produced during system startup. Look for a message similar to the one below in the command output. This message gives the amount of memory in your system in bytes.

```
Memory Information:
    Physical:  98304 KBytes ...
```

In addition to real memory requirements, there is a minimum system swap space size requirement that is described below.

Disk Space

It requires approximately 70 megabytes of disk storage to install either version of AVS 5.5 from the CD. More disk storage may be required for swap space (see below).

Swap Space

Recommended Swap Space Size

We recommend that swap space be a minimum of twice the amount of main memory or 64 megabytes, whichever is greater. Very large datasets and complex networks may require more than the minimum recommended swap space.

Determining How Much Swap Space You Have

To determine the current amount of swap space available on your system, log in as *root* and type the following command:

```
/etc/swapinfo -mt
```

This produces a listing like the following:

```
             Mb       Mb       Mb   PCT  START/       Mb
TYPE      AVAIL     USED     FREE  USED   LIMIT RESERVE  PRI  NAME
dev         512        3      509    1%       0       -    1  /dev/vg00/lvol2
reserve       -       39      -39
memory      188      130       58   69%
total       700      172      528   25%       -       0    -
```

The last line lists the totals across all swap partitions. This system has over 700 meg of swap space.

Increasing System Swap Space

If the amount of swap space currently available on your system is inadequate, follow the following procedure to augment the existing swap space with file system swap space.

1. Log in as *root* and invoke the System Administration Manager by typing:

   ```
   sam
   ```

2. Select the entry labeled *Disk and File Systems* -> from the main menu. Then select the *Swap* -> entry. A panel appears listing the mount directories and their "Total Kbytes."

```
Device File/                  Total      Current     Enable  Enable
Mount Directory     Type      Kbytes     Priority    Now     on Boot

/dev/dsk/c201d5so   dev        71766     0           y       y
/dev/dsk/c201d6so   dev       101855     0           y       n
```

3. Select *Add File System Swap* under the *Actions* menu. A new panel appears listing the "Unused Kbytes" and the "Total Kbytes," as well as the mount of point for the disks.

4. Edit the *Maximum Swap* typein to increase the amount of file system swap in your system.

5. Press the *OK* button to execute your request.

Software Prerequisites

HPUX

AVS 5.5 will run on the HP-UX 10.20 operating system; it will not run on earlier versions. This platform appears as *HP1020* in the installation menu.

The *uname -r* command will show the current revision of HP-UX that is running on your HP workstation.

OpenGL

The default kernel, *avs*, requires that the OpenGL libraries be installed on the target machine. AVS was built using OpenGL 1.1. You can determine which version of OpenGL you have by typing the following as root user:

```
/usr/sbin/swlist | grep OpenGL
```

```
B6196AA_APZ B.10.20.01 HP-UX 700 OpenGL 3D Graphics API Run Time Environment
PHSS_15838  B.10.00.00.AA OpenGL 1.1 Revision 1.07 Developers patch
PHSS_15839  B.10.00.00.AA OpenGL 1.1 Revision 1.07 Runtime patch
```

End users of AVS require the Run Time Environment to use the OpenGL kernel.

Older systems may not be OpenGL compatible or perform as well with OpenGL as they do with the PHIGS interface. If you don’t have OpenGL installed check with Hewlett Packard to determine what is the best interface for your configuration. If you have neither PHIGS nor OpenGL installed, use the *avs.phigs* kernel and the dummy libraries noted below with "-nohw".

PHIGS and PowerShade

Hardware Rendering: PHIGS and PowerShade Required

The *avs.phigs* renderer supports hardware rendering on the graphics adapters mentioned previously. In order to access this funtionality, AVS requires that version 3.40 or later for HP-UX 10.20. PowerShade must also be installed on the workstation.

You can determine which version of PHIGS you have by typing the following as root user:

```
/usr/sbin/swlist | grep PHIGS
```

These products are not included with base HP workstations and may need to be ordered separately. However, all of the "Z" adapters (e.g., CRX24Z, CRX48Z, HCRX-8Z, etc) have PowerShade built in. To verify that PowerShade is available on your system, select "General" from the "Toolboxes" pop-up menu on the HP VUE panel, and then select "System Info".

If the PHIGS library is not present and hardware rendering is enabled in *avs.phigs*, the following message will appear when AVS is started:

```
/lib/dld.sl: Can’t open shared library: /opt/graphics/phigs/lib/libphigs.sl
/lib/dld.sl: No such file or directory
Terminated
```

Software Rendering: Dummy PHIGS Library Required

If you do not have the PHIGS runtime library and PowerShade, AVS will still work with the software renderer since it does not make any PHIGS library calls.

However, prior to running AVS using the software renderer, you need to supply a dummy library that will masquerade as the PHIGS library to satisfy the dynamic loader at AVS startup. We supply such a dummy library in *<install-dir>/lib/libphigs.sl*.

Login as root and copy:

```
mkdir -p /opt/graphics/phigs/lib
cp  $AVS_PATH/lib/libphigs.sl  /opt/graphics/phigs/lib/libphigs.1
ln -s /opt/graphics/phigs/lib/libphigs.1 /opt/graphics/phigs/lib/libphigs.sl
```

Then, make *libphigs.1* world readable and executable.

In this case, you should start AVS using the *-nohw* command line option or include the **NoHW** keyword in your personal *.avsrc* file. If you don't do this, you may see a cryptic message about not being able to find the "popph" function.

Error Message: Invalid version for shared library

The revision date of the */opt/graphics/phigs/lib/libphigs.sl* library must be greater than or equal to the date of the PHIGS library AVS Inc. used to link the AVS application. If you somehow run afoul of this limitation, this is the error message you will see:

```
/lib/dld.sl:  Invalid version for shared library:  /usr/lib/libphigs.sl
/lib/dld.sl:  Exec format error
IOT trap
```

and AVS will exit.

Using AVS on an HP 9000/7*xx* Workstation

The following sections describe in detail the differences between using AVS on an HP workstation and AVS as it is described in the standard documentation.

Peripherals

Dial Box

The Dial Box I/O device mentioned in the *AVS User's Guide* is not supported in this release of this product.

Spaceball

The Spaceball device is supported.

Colormap Sharing

Colormap sharing can be a problem if you are using a pseudo color visual type and have hardware rendering enabled. This affects the CRX graphics adapter and the built-in graphics adapters.

There are several steps that need to be taken in order to have AVS, the HP X server and OpenGL or PHIGS share the same colormap.

For Both Hardware and Software Rendering

By default, HP VUE will allocate about forty colors from the default colormap that AVS will try to share. If a message similar to the following appears when AVS starts up, it means that AVS was forced to use less colors than is optimal:

```
Warning: reducing color usage: R=5,G=5,B=5,Grey=17
```

In order to reduce VUE's color usage, so AVS can acquire all the colors that it would like, follow this selection sequence:

```
Style Manager
  Color
     HPVue Color Use
        Low Color
           OK
```

Then, logout and login again.

Next, you must also use the Edit Resources action to add the following line to your resource database:

```
*shadowPixmaps   True
```

To pick Edit Resources, follow this selection sequence:

```
Toolboxes
  General
    System_Admin
      Edit Resources
```

In order to have these resources take effect properly, all clients (e.g. *hpterms*) must be killed and you must logout and log back in.

Hardware Rendering Only

There is an environment variable that needs to be set to tell applications to share the existing colormap, if possible, instead of defining their own. The following line should be inserted in the X configuration file */usr/lib/X11/vue/Vuelogin/Xconfig*.

```
Vuelogin*environment:   SB_X_SHARED_CMAP=true
```

After doing this, the X server needs to be restarted so the variable will take effect. The easiest way to restart the X server is to select *Restart Server* from the *Options* menu on the logon screen. See the chapter "The HP A1659A (CRX) and HP A1924A (GRX) Devices" in the *HP PHIGS Workstation Characteristics and Implementation* manual for an in depth discussion of this environment variable.

All Rendering

Finally, AVS needs to have its internal gamma correction disabled so it can share the colormap with X and OpenGL or PHIGS. This can be done by using the *-gamma* command line option or the **Gamma** keyword in your personal *.avsrc*. In either case, the variable should be set to *1.0*.

A side of effect of disabling AVS's gamma correction is that the user interface in AVS becomes a little dim. To correct this, start AVS with the *-class X* command line option or edit the *avs.Xdefaults* file as described in the "AVS on Color X Servers" appendix of the *AVS User's Guide*.

Enabling Double Buffering on Builtin Graphics Boards

If you have an HP system with a builtin graphics board (e.g., 705, 710, 715, 725, etc.) and you want to use hardware rendering, you will want to enable double buffering. This will give you access to full 8/8 double buffering using virtual memory.

Set the following environment variable to enable this feature:

```
setenv HP_VM_DOUBLE_BUFFER  TRUE
```

There are two tradeoffs that should be considered when enabling this feature: speed and memory usage. Virtual memory rendering uses pure software algorithms. As a result, rendering can be significantly slower. Virtual memory rendering also consumes a part of memory that would otherwise be available to the application. As a result, you should carefully consider whether it might just be more effective to use AVS's software renderer.

Starting AVS

The only issues here that differ from the account in the AVS *User's Guide* and the *avs* man page are the default selection of the software renderer and selecting the correct X server VisualType for your graphics adapter.

This involves making changes to your personal AVS startup file (*.avsrc*) in your HOME directory. If you don't already have a *.avsrc* file, copy the sample file from *<installdir>/avs/runtime/avsrc* to your home directory.

Renderers

The default kernel *avs* requires the OpenGL runtime libraries to be present to run. The *avs.phigs* renderer requires that PHIGS and PowerShade be present on the workstation for hardware rendering to work. These packages are not part of the base HP configuration and must be purchased from HP separately; AVS enables hardware rendering by default so if you do not have PHIGS available you need to use the dummy library and set NoHW on as follows. This is done with the following line in the AVS startup file that is located in *<installdir>/avs/runtime/avsrc*.

```
NoHW    1
```

If you have both PHIGS and PowerShade on your workstation and would like to use hardware rendering, they should work by default. **Note:** AVS 5.4 used NoHW as the default; AVS 5.5 enables hardware rendering.

Visual Type

No matter what X server visuals are available, AVS will always use the default visual. The default visual on all the supported HP graphics adapters is pseudo color. However, if you have one of the 24-bit graphics adapters (CRX24, CRX24Z, CRX48Z, TVRX, HCRX-24, HCRX-24Z) you will want to use either a 24-bit direct color visual or true color visual.

To see which 24-bit visuals exist for your display, type the command:

```
/usr/contrib/bin/X11/xdpyinfo
```

Note: this standard X utility may not be available on your HP.

To have AVS use a 24-bit direct color visual or true color visual that these graphics adapters support, add this line in your personal *.avsrc* file or use the new *-vistype* command line option documented above in the "Unexposed command line options" section.

```
VisualType      DirectColor  (or TrueColor)
```

If you do not, AVS will use a visual type that is only 8-bits when the graphics adapter is able to support a 24-bit visual.

Image Viewer

All functions and behavior in the Image Viewer are as described in the AVS *User's Guide*. The Image Viewer will use as many bits of color as are available on the graphics adapter.

Network Editor

The Network Editor functions as described in the AVS *User's Guide* and *Module Reference* manual, with these few exceptions:

Modules

The following modules documented in the *AVS Module Reference* manual do not appear in the AVS release for HP workstation because they require rendering features that are not supported in this release:

alpha blend (requires hardware-assisted alpha transparency)
transform pixmap (requires 2D texture mapping)

Remote Module Execution

The command to establish contact with a remote host to execute remote modules is different on HP workstations. Instead of using *rsh* as on most other systems, use *remsh* in the *hosts* file.

Image Output

While using the hardware renderer, if you are using the **image to postscript** module to get output from the **geometry viewer** module, or any of the Animation Application output modules such as **write frame seq**, all parts of the Geometry Viewer scene window must be visible. No parts of the display window can be off the screen or obscured by other windows. If they are, any module downstream of **geometry viewer** will not receive the off-screen or obscured portion of the pixmap.

Graph Viewer

Other than the fact that image data loaded as a background to a plot on 8-bit graphics adapters will be dithered to pseudocolor rather than appear in true color, there are no differences between AVS Graph Viewer behavior on HP workstations and the Graph Viewer descriptions in the AVS documentation.

Geometry Viewer

The Geometry Viewer (or the **geometry viewer** module) will be using either the software renderer or the hardware renderer to produce the contents of its scene windows.

Stereo Support

The default OpenGL *avs* kernel now includes basic support for stereo viewing using quad-buffered "stereo-in-a-window". The stereo viewing device supported is the CrystalEyes stereo goggles from StereoGraphics. See the "AVS 5.5 on UNIX Platforms" chapter for more general information.

HP stereo is only supported on the Visualize-FX4 and Visualize-FX6 graphics adapters. Stereo needs to be enabled as the root user using either the *SAM* System Administrator Utility or *setmon* or by making changes in the Boot ROM. More information on enabling stereo and selecting the right hardware and cables can be found in the following whitepaper on the HP web site:

http://www.hp.com/visualize/support/library/stereographics.pdf.

In order to enable stereo you need to run the *setmon* utility as follows:

```
/opt/graphics/common/bin/setmon
```

Select one of the "Stereo in a Window" monitor entries, usually 4 or 5 depending on your system. The utility will switch over to the new entry for 15 seconds to see if it is working and if you don't confirm it, it will revert to the original setting. It will automatically reset the X server so existing applications will be closed.

Once selected, you can enable stereo using the Geom Viewer/Camera/Stereo button or Control-Mouse-Left in the view window. If stereo is not enabled, enabled stereo will cause a shift in the view but will not cause any problems.

Rendering Features

The following table lists the rendering features described in the "Geometry Viewer" chapter of the *AVS User's Guide*, including new AVS features, down the left column, and the AVS software renderer and the AVS hardware renderer's PHIGS and OpenGL implementations across the top. The table intersections show which features are present on each platform, and draw your attention to more detailed explanations of behavior later in this section.

Table 5-1. Geometry Viewer Behavior

Rendering Feature	Software Renderer	PHIGS	OpenGL
Arbitrary Clip Planes	yes, 8 planes	no (note 1)	no (note 1)
Geometric			
Volume Rendering	yes (note 2)	no (note 2)	no (note 2)
Vertex Transparency	no (note 3)	no (note 3)	yes (note 3)
Vertex Colors	yes (note 4)	yes	yes
Edit Property			
RGB/HSV Colors	yes (note 4)	yes	yes
Ambient	yes	yes	yes
Diffusion	yes	yes	yes
Specular Highlights	yes	yes	yes
Gloss	yes	yes	yes
Transparency	yes (note 5)	yes (note 6)	yes (note 6)
Metallic	yes	yes	yes
Edit Texture	yes	no	yes
2D Texture Mapping	yes	no	yes
3D Texture Mapping	yes	no	yes
Filtered Textures	no	no	yes
Tile Textures	no	no	no
Alpha Textures	yes	no	no
Rendering Options			
Points	yes	yes	yes
Lines	yes	yes	yes
Smooth Lines	no	no	no
No Lighting	yes	yes	yes
Flat Shading	yes	yes	yes
Outline Gouraud	yes	yes (note 10)	yes
Gouraud Shading	yes	yes	yes
Phong Shading	no	no	no
Backface Properties			
Cull front	no	yes	no
Cull back	yes	yes	yes
Flip normals	no	no	no
Lights			
Number of Sources	16	16	8
Ambient	yes	yes	yes
Directional	yes	yes	yes
Bi-Directional	yes	yes	yes
Point	no	yes	yes
Spot	no	yes	no
Light Colors	yes	yes	yes

Table 5-2. Geometry Viewer Behavior Across Platforms (continued)

Rendering Feature	Software Renderer	PHIGS	OpenGL
Cameras			
Depth Cue	yes	yes	yes
Global Anti-Alias	no	yes (note 7)	yes
Accelerate	no	no	no
Perspective	yes	yes	yes
Axes for Scene	yes	yes	yes
Front/Back Clipping	yes	yes	yes
Sorted Transparency	no	no	no
Polygonal Spheres	yes (note 8)	no (note 9)	no
Double Buffer	no	no	yes
Stereo	no	no	yes (note 11)
Labels			
Drop Shadow	no	no	yes
Title	yes	yes	yes
Stroke	no	yes	yes
Kanji	yes	no	no

Notes

1. Arbitrary clipping planes are used by the **clip geom** module.

2. "Volume Rendering" is here defined in a very narrow sense: support of a specific option to the **GEOMedit_texture_options** call that is used by the **volume render** module. The fact that the hardware renderer does not support this form of volume rendering only means that it cannot use one of the features of the **volume render** module; it does not mean that the hardware renderer cannot render volumes of data using other rendering techniques.

3. Vertex transparency is used by the **colorize geom** module.

4. When using the software renderer, the number of colors that will appear on the screen (216 pseudocolor, 16,777,216 true color, or some number in between) is dependent upon the X server visual support present on the *display* hardware. Internally, 24-bit true color is always used and will be output on the **geometry viewer** module's image output port.

5. The software renderer supports single-pass transparency. Single-pass transparency correctly renders a transparent surface that is over an opaque surface. (Multi-pass transparency is required to correctly render multiple overlapping transparent surfaces.)

6. Screen door transparency is supported through PHIGS when hardware rendering is enabled in the *avs.phigs* renderer.

7. Global Anti-Aliasing of lines is supported in PHIGS. This is almost the same as the **Smooth Lines** rendering option, but it is on a per view basis instead of a per object basis.

8. A feature in the software renderer is the ability to render spheres directly instead of subdividing them into polygons. This saves a tremendous amount of memory when rendering spheres. The algorithm's execution speed is bound by the size of the spheres it has to render. This feature may be disabled by toggling the **Polygonal Spheres** button under the Cameras menu. Spheres will now be rendered by subdividing them into polygons.

9. Spheres are subdivided into polygons when using the PHIGS based hardware renderer.

10. Turn on **Front/Back Clipping** to remove rendering artifacts.

11. Stereo support requires Visualize-FX4 or Visualize-FX6

Programming Considerations

AVS, its modules, and component libraries were compiled using the HP-UX C compiler (10.32.05) and the standard Unix C libraries located in */lib/libc*. Its FORTRAN components were compiled using the HP-UX FORTRAN 77 compiler (10.20.09)

Note: Please see the information on "GEOMint_color" in Chapter 3, *AVS5.5 on UNIX Platforms* for an issue that will affect modules using the GEOM library on 64 bit platforms.

Module Generator

The AVS Module Generator automatically invokes an *xterm* window when you press **Compile** to compile a module. However, the */usr/bin/X11* directory that contains the *xterm* command is not in the default path given to new users on HP-UX. To use the Module Generator, you should add */usr/bin/X11* to your default path.

csh users would add a line like the following to one of their startup files, usually *.cshrc* or *.login*:

```
set path=($path /usr/bin/X11)
```

sh or *ksh* users would add a line like the following to one of their startup files, usually *.profile*:

```
PATH=$PATH:/usr/bin/X11
```

Compiling and Linking Modules

Use the Example Makefile as Template

You should use the makefile in *<installdir>/avs/examples/Makefile* as the template for your own FORTRAN and C module makefiles. This makefile in turn includes the file *<installdir>/avs/include/Makeinclude* that contains additional macro definitions appropriate to the HP platform. These include references to the libraries to be found in */opt/graphics* (see definitions for *PHIGSINC* and *PHIGSLIB* in Makeinclude).

Fortran

The way in which the offset returned by the field accessor routines, **AVSptr_offset** and **AVSfield_points_offset**, is used can cause segmentation faults. This is due to the fact that on the HP PA-RISC architecture, the address space is actually divided into four segments. Depending on the instruction being executed, a test may be done to see if a segment is being crossed. This can happen in AVS because the Fortran accessor routines were written in order to be portable to different platforms. See bug **X7787** in the section "AVS on HP Workstations: Known Problems" for a description and a workaround.

All of the Fortran code in AVS was compiled using the *+ppu* switch. This causes an underscore to be postpended to all definitions and references to externally visible symbols. This may be an issue if you are trying to use an existing library of Fortran routines without recompiling.

AVS on HP Workstations: Known Problems

This is the list of known problems for AVS 5.5 on the HP 9000/7*xx* platforms.

X7787: Failure using field accessor routines in Fortran

Problem Description:

The example *threshold_f.f* fails when trying to use the offset returned by **AVSptr_offset**. The same behavior is exhibited by the routine **AVSfield_points_offset**.

Workaround: Refer to the documentation in the *AVS Developer's Guide* on **AVSptr_offset.** Of the three ways illustrated to use the offset returned by the accessor routine, only the second fails. As a workaround, call a routine passing it the address returned by the offset routine. Once inside this routine, the data can be accessed correctly.

X9736: Transparency not working on CRX24Z, CRX48Z, and TVRX

Problem Description:

avs.phigs supports all the HP platforms using PHIGS. While this provides a consistent interface across the graphics adapters, it does not allow access to some of the functionality on all graphics adapters. Alpha transparency is an example of this on the CRX48Z, CRX24Z and TVRX.

The PHIGS API only allows AVS to access screen door transparency. As a result, depending on the transparency level specified, a more or less apparent "screen door" pattern appears on the geometry.

X10207: HP CRX48Z objects get dimmer as become smaller

Problem Description:

The CRX48Z and CRX24Z sometimes exhibit a problem where lighting seems to get turned off as the object is made smaller. Lighting comes back on as the object is made larger again. A problem report has been filed with HP but no fix for the problem is available at this time.

X10640: Hardware rendering dependent on shared PHIGS library

Problem Description:

The PHIGS kernel *avs.phigs* depends on finding a compatible copy of the shared PHIGS library (*libphigs.sl*) in */usr/lib* even when the hardware renderer is not being used. If AVS does not find any shared PHIGS library it will print this error message and exit:

```
/lib/dld.sl: Can’t open shared library: /usr/lib/libphigs.sl
/lib/dld.sl: No such file or directory
IOT trap
```

Workaround: This is covered in the "System Prerequisites" section above.

Problem: AVS/Graph statically linked

Problem Description:

The HP-UX 10.20 platform uses Toolmaster 7.1 for AVS/Graph support. During development of AVS 5.5, it was necessary to statically link in *libhpgfx.a* because of linker problems. Let support know if you have any trouble on different configurations for any reason (support@avs.com).

17547: HP: AVSGraph module may crash on exit

Problem Description:

In some HP-UX 10.20 configurations the AVSGraph module crashes when exiting, leaving a core file and listing a traceback; on other machines, it exits cleanly. This is still under investigation. It does not impact the utility of the module; repeated crashes will overwrite the same core file.