AVS 5.5 FOR SILICON GRAPHICS

CHAPTER SEVEN

Introduction

This chapter describes AVS 5.5 as it runs on Silicon Graphics Incorporated workstations. There are two different versions of AVS 5.5 for SGI: one for 32-bit architectures (N32) and one for 64-bit architectures (N64); both versions are addressed in this chapter. 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: The supported operating system has been updated to IRIX 6.5 for both 32 bit and 64 bit architectures. The O32 platform has been dropped and the N32/Mips3 has been added. The N64/Mips4 platform has been updated from IRIX 6.2 to IRIX 6.5. Stereo support has been reworked to provide quad-buffered (stereo-in-a-window) support in addition to full screen stereo, borrowing from the AVS/Express 5.0 implementation. Several SGI specific bugs have been addressed. No other significant platform specific changes have been made.

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 any workstation in the Silicon Graphics 4D product line, including the IRIS Indigo, Crimson, Challenge, Onyx, Octane and O2. There are two versions of AVS 5.5 for Silicon Graphics (SGI) platforms:

SGN32 supports 32-bit architectures:
This version executes on any SGI workstation or server using the 32-bit processors running IRIX 6.5 or above, and may be used on 64-bit

processors if necessary. The executables use the "n32" (new 32 bit) binary format and can be run on "n32" or "n64" platforms; the code was compiled for Mips3 compatible architectures. Products that use this architecture include Indy, Indigo, Crimson, Challenge, Onyx, and O2, running a version of IRIX 6.5. OpenGL support only is provided; there is no longer any GL renderer due to SGI no longer supporting this interface in IRIX 6.5.

SGN64 requires 64-bit architectures:
This version executes on any SGI workstation or server using a 64-bit processor. The executables use the "n64" (new 64 bit) binary format and can only run on 64 bit platforms; the code was compiled for Mips4 compatible architectures. Products that use this architecture include POWER Indigo 2, POWER Challenge, and Octane, running a version of the IRIX 6.5 operating system or later.

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 standard graphics library; 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.

NOTE: Both SGN32 and SGN64 now use OpenGL for hardware rendering.

The **software renderer** will work on any SGI workstation with an 8- or 24-plane color frame buffer, the XPutImage *xlib* call, and a local X server that supports them.

In principle, the **hardware renderer** should work on any SGI workstation equipped with an 8- or 24-plane display, and a graphics subsystem—either hardware, or software-emulated (as on the Indigo)—that supports the standard graphics library. However, in addition, the graphics subsystem *must* support a Z-buffer. If your system does not support a Z-buffer, then you will need to use the software renderer exclusively.

See the "Geometry Viewer" section below for specific information on which rendering features are supported by the software and hardware renderers.

Memory Requirements

AVS requires that the system upon which it will execute be configured with a minimum of 16 megabytes of main memory. More real memory, such as 32 or 64 megabytes, is strongly recommended.

Performance will be improved if there is more real memory. In addition to real memory requirements, there is a minimum system swap space requirement (see below).

Disk Space

AVS 5.5 requires approximately 87 megabytes (SGN32 32-bit) or 90 megabytes (SGN64 64-bit) of disk storage to install AVS from the CD. More disk storage may be required for swap space (see below). The *df -k* command reports available disk space.

Determining Your Configuration

To interactively determine your SGI workstation's hardware configuration, type the */bin/hinv* command. This produces a listing like the following:

```
1 30 MHZ IP12 Processor
FPU: MIPS R2010A/R3010 VLSI Floating Point Chip Revision: 4.0
CPU: MIPS R2000A/R3000 Processor Chip Revision: 3.0
On-board serial ports: 4
Data cache size: 64 Kbytes
Instruction cache size: 64 Kbytes
Main memory size: 32 Mbytes
Integral Ethernet: ec0, version 0
Tape drive: unit 2 on SCSI controller 0: QIC 150
Disk drive: unit 1 on SCSI controller 0
Integral SCSI controller 0: Version WD33C93A
Graphics board: GR1.2 Bit-plane, Z-buffer, Turbo options installed
```

The "Main memory size" line reports the memory size. The "Graphics board" line reports the hardware graphics present.

For more information on graphics hardware options, such as the number of texture memory modules installed, use the */usr/gfx/gfxinfo* command. This produces a listing like this:

```
Graphics board 0 is "IMPACT" graphics.
   Managed ("0:0") 1280x1024
   Product ID 0x0, 1 GE, 1 RE, 1 TRAM
   MGRAS revision 3, RA revision 5
   HQ rev A, GE11 rev B, RE4 rev A, PP1 rev A,
   VC3 rev A, CMAP rev D, MC rev C
   unknown, assuming 19" monitor (id 0x9)
```

Software Prerequisites

IRIX Operating System

There are two different versions of AVS for SGI.

These products appear as different options during installation on multi-product media:

SGN32
AVS for IRIX release 6.5 or later. Requires that 6.5 or later of the IRIX operating system and its accompanying libraries and compilers be installed on the system upon which AVS will execute. This version was compiled on an R10000 with the Mips3 flag.

SGN64
AVS for IRIX release 6.5 or later. Requires that 6.5 or later of the IRIX operating system and its accompanying libraries and compilers be installed on the system upon which AVS will execute. This version was compiled on the R10000 using the Mips4 flag and will not run on R3000 or R4000 machines.

The differences between the SGN32 and the SGN64 AVS versions are:

- The SGN32 libraries are the "N32" Application Binary Interface (ABI), whereas the SGN64 libraries are in the "n64" ABI. A older format, the "o32" ABI, is no longer supported.

The */bin/uname -a* command will list which version of IRIX is running on the system.

Swap Space

Recommended Swap Space Size

We recommend that your system be configured with a minimum swap space size that is twice the real memory size or 64 megabytes, whichever is *greater*. Very large datasets flowing through networks with many modules and connections may require even larger swap spaces.

Determining How Much Swap Space You Have

Most SGI workstations come with the default system swap space size set to about 50 megabytes. To see how much swap space is available to your system, type the */etc/swap -l* command as follows:

```
/etc/swap -l
path              dev  swaplo blocks   free
/dev/dsk/dks0d1s1 22,33      0  81728  62000
```

The "blocks" column is the total swap space reserved on the system in 512-byte blocks, both in the system swap space partition and any additional swap space that might have been defined for the system on other partitions; "free" is the amount currently available for allocation and will vary with the load on the system.

Running out of Swap Space: Message, Abort, or Hang

Depending upon the circumstances, when AVS runs out of swap space it either produces a message box, aborts, or hangs. In many cases, AVS itself receives the error during a *malloc* call. In this instance, either individual modules or the entire AVS system may abort with the following error message:

```
Failure allocating memory:
See Installation/Release Notes to increase swap space and/or
shared memory segment size
```

This message may be followed by many additional "tcp read" error messages as each module expires. Scroll to the top of the error messages to see this first message that indicates the actual cause of the failure.

In some cases, other parts of the system will generate the error and you will not see the above message. If you suspect that swap space may be the culprit, you can restart AVS, repeat the sequence of steps that led to the abort or hang, while monitoring AVS's consumption of swap space by entering successive */etc/swap -l* commands.

How to Increase System Swap Space

See the Silicon Graphics *IRIX System Administrator's Guide* for information on how to increase the system swap space size.

Using AVS on an SGI Workstation

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

General

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. See bug report **X8581** in the "Known Problems" section below for information on how to correctly activate the Spaceball.

Window Manager

AVS works with any of the standard X Window System window managers, including *4Dwm*, SGI's implementation of a Motif-style window manager. No user modifications are necessary in order to use AVS with *4Dwm*. However, *4Dwm*'s interaction norms do differ somewhat from the earlier X window managers under which AVS was originally developed. See the "Window Manager" section in Chapter 3, *AVS 5.5 on UNIX Platforms*.

Starting AVS

There are four differences to starting AVS on an SGI workstation from the account in the AVS *User's Guide* and the *avs* man page. These are

- changing your display size if you have a small monitor.
- setting the correct X server VisualType for your graphics adapter (if necessary).
- setting a gamma correction factor (if necessary).
- setting a value for the **AVS_XOR_VALUE** environment variable to make certain display lines, such as that used for the Bounding Box, visible. This is only necessary on some graphics adapters, e.g., the Crimson with Elan graphics.

Reconfiguring the first three items listed 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 and edit it to include the necessary lines. **AVS_XOR_VALUE** is an environment variable that must be set through your system's shell.

Display Size

AVS normally expects to execute on 1280x1024 resolution displays. If the display is smaller or larger than this, AVS will try to automatically rescale its interface to fit on the screen. If AVS's automatic efforts are not satisfactory (for example, the bottoms of menus are being clipped by the window manager or the Network Editor panel is overlapping other windows), you can explicitly set a virtual screen size and a position for the Network Editor window by adding lines such as the following to your personal *.avsrc* file or using the *-size* command line option:

```
NetworkWindow       850x850+250+20
ScreenSize          1080x864
```

Visual Type: Seeing Pseudocolor on a Truecolor Adapter

This issue affects some 24-plane true color graphics systems.

IRIX often starts the X Window System server using a pseudo color X visual (as reported by the *xdpyinfo* command) as its default visual, even though the display hardware is capable of supporting 24-plane true color. By default, AVS always uses this default X visual when it starts.

On 8-plane pseudo color systems, this is fine. AVS will use the correct pseudo color visual for the 8-bit platform.

However, on 24-plane graphics systems, AVS will also come up using the 8-bit default pseudo color visual, unnecessarily limiting the color resolution.

Permanently Changing the Default Visual Systemwide. You can permanently change the X Window System's default visual.

1. As root, edit the file */usr/lib/X11/xdm/Xservers*.

2. Change the line:

   ```
   :0 secure /usr/bin/X11/X -bs -c -pseudomap 4sight
   ```

   To read:

   ```
   :0 secure /usr/bin/X11/X -class TrueColor -depth 24 -bs -c -pseudomap 4sight
   ```

3. Kill and restart the X server.

SGI Crimson, Invisible Bounding Boxes, and AVS_XOR_VALUE Environment Variable

On a few SGI platforms, such as the SGI Crimson with Elan graphics, the "xor'd" pixel value used to draw lines such as the Geometry Viewer's Bounding Box are invisible.

To make these lines visible, you must set the AVS_XOR_VALUE environment variable in one of your startup files (usually *.login* or *.csh* for *csh*, and *.profile* for *sh* or *ksh*). Set the environment variable as follows:

```
csh:          setenv AVS_XOR_VALUE 0xnnnnnn

sh or ksh:    AVS_XOR_VALUE=0xnnnnnn; export AVS_XOR_VALUE
```

0x*nnnnnn* is a pixel value. The first two characters are a zero (not an "oh") and the lowercase character "x". This is a 24-bit number, with 1 byte for red, 1 byte for green, and 1 byte for blue. The *nnnnnn* is thus two hexadecimal characters each (range 0 through f) for red, green, and blue. On truecolor systems, all three bytes are used to determine the line color. On pseudocolor systems, only the low order byte is significant. It is used as an index into the hardware colormap.

You will need to experiment (e.g., 0x8a8a8a, 0xa2a2a2, etc.) to determine a good combination for your platform. On truecolor systems, equal values for red, green, and blue produce a whitish line. (0xaaaaaa is the default AVS value that produces the nearly invisible lines.) On pseudocolor systems, since the lower order byte is a colormap index, it is hard to predict the result of any given number.

Image Viewer

All functions and behavior in the Image Viewer are as described in the AVS *User's Guide*. Color rendition will be up to the capabilities of the X server.

On pseudocolor systems, the Image Viewer's **Images** submenu will have a choice of dithering options.

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 SGI workstations.

alpha blend
transform pixmap

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

There are no significant differences between AVS Graph Viewer behavior on SGI workstations and the Graph Viewer descriptions in the AVS documentation. On pseudo color systems image data loaded as a background to a plot will be dithered to 8-bit pseudo color rather than appear in true color.

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.

Color

The OpenGL graphics subsystem that produces the contents of Geometry Viewer scene windows under the hardware renderer uses its own visuals (as distinct from X Window System visuals). On some systems and visual types, OpenGL may split the number of planes in two in order to support "double buffering". For example on 24-plane systems, it may use 12-planes for each buffer, providing 4096 colors instead of the 16 million supported by the full 24 planes.

"Double buffering" just means that OpenGL renders its output image into an off-screen pixmap, then transfers the image to the screen, switching back and forth between the two buffers. "Single buffering" means that OpenGL renders its output image directly to the screen. If single buffer mode is set, you can watch the object being drawn; if double buffer mode is set, the completed picture simply appears in the window.

To see a full 24-plane true color rendering, you might have to change to single buffer mode by switching off the **Double Buffer** button on the **Cameras** submenu. There is, at present, no AVS startup option that will automatically switch off double buffering.

Stereo Support

Both versions of AVS 5.5 include basic support for stereo viewing using OpenGL. 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.

Stereo is enabled using either the *setmon* command or the *ircombine* command depending on the type of machine (higher end graphics adapters like the Infinite Reality cards usually use *ircombine* though they may also still support *setmon*).

If */usr/gfx/ircombine* exists, it is used automatically with the following settings to enable and disable stereo:

```
on  - /usr/gfx/ircombine -destination active -channel 0 format=1024x768_96s.vfo
off - /usr/gfx/ircombine -destination active -channel 0 format=1280x1024_72.vfo
```

Otherwise if */usr/gfx/setmon* exists, it is used with the following settings:

For Reality Engine:

```
on - /usr/gfx/setmon -n 1025x768_96s
off - /usr/gfx/setmon -n 1280x1024_60
```

For VGX and VTX:

```
on - /usr/gfx/setmon -n STR_BOT
off - /usr/gfx/setmon -n 60HZ
```

All others:

```
on - /usr/gfx/setmon -n STR_BOT
off - /usr/gfx/setmon -n 72HZ
```

You can see the current setting using two debug environment flags:

```
setenv AVS_STEREO_CMDS_ARE 1
setenv AVS_OGL_INFO 1
```

In order to override these values, there are two environment variables to set the "on" and "off" commands:

- **AVS_STEREO_ON_CMD** specifies the command that enables stereo mode

- **AVS_STEREO_OFF_CMD** specifies the command that disables stereo mode

Because AVS does NOT "remember" your previous mode before it entered stereo, you may want to at least set the "off" variable, AVS_STEREO_OFF_CMD.

To exit stereo mode, type **Ctrl-left mouse button** again or use the Camera/Stereo toggle button. **Note:** You should exit stereo mode before leaving AVS or it may leave stereo enabled. If you do leave stereo on by accident, restart AVS and toggle stereo on/off to leave stereo mode; or use the "off" stereo command shown above to exit stereo manually.

For information on known problems wtih SGI stereo, see the Known Problems section at the end of this chapter.

Rendering Features

The following table lists the rendering features described in the "Geometry Viewer" chapter of the *AVS User's Guide*, including new AVS 5 features, down the left column, and the AVS software renderer and the AVS hardware renderer's OpenGL implementation 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 7-1. Geometry Viewer Behavior Across Platforms

Rendering Feature	Software Renderer	Hardware (OpenGL)
Arbitrary Clip Planes	yes, 8 planes	no (note 1)
Geometric		
Volume Rendering	yes (note 2)	yes (note 2)
Vertex Transparency	no (note 3)	yes (note 3)
Vertex Colors	yes (note 4)	yes
Edit Property		
RGB/HSV Colors	yes (note 4)	yes
Ambient	yes	yes (note 5)
Diffuse Lighting	yes	yes (note 5)
Specular Highlights	yes	yes
Gloss	yes	yes
Transparency	yes (note 6)	yes (note 7)
Metallic	yes	yes
Edit Texture	yes	yes (note 7)
2D Texture Mapping	yes	yes
3D Texture Mapping	yes	yes (note 2)
Filtered Textures	no	yes
Alpha Textures	yes	no
Rendering Options		
Points	yes	yes
Lines	yes	yes
Smooth Lines	no	no
No Lighting	yes	yes
Flat Shading	yes	yes
Gouraud Shading	yes	yes
Outline Gouraud	yes	yes
Phong Shading	no	no
Backface Properties		
Cull front	no	no
Cull back	yes	yes
Flip normals	no	no
Lights		
Number of Sources	16/8 (note 8)	8
Ambient	yes	yes
Directional	yes	yes
Bi-Directional	yes	yes
Point	no	yes
Spot	no	no
Light Colors	yes	yes

Table 7-2. Geometry Viewer Behavior Across Platforms (continued)

Rendering Feature	Software Renderer	Hardware (OpenGL)
Cameras		
Depth Cue	yes	yes
Perspective	yes	yes
Accelerate	no	no
Front/Back Clipping	yes	yes
Axes for Scene	yes	yes
Double Buffer	no	yes
Sorted Transparency	no	no
Shadows	no	no
Polygonal Spheres	yes	no
Stereo	no	yes
Head Tracking	no	no
Labels		
Drop Shadow	no	yes
Stroke Text	no	no
Kanji	yes	no

Notes

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

2. Currently, only the very high-end platforms (Reality Engine and Onyx) support this feature.

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 pseudo color, 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 **Ambient** and **Diffuse** sliders have no effect on objects with vertex colors.

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

7. Transparency and texture mapping are supported on those SGI systems where these capabilities are available. On systems that do

not have support for them, enabling the feature has no effect.

8. When, at AVS startup, you initialize the software renderer only (*-nohw* or **NoHW**), then there will be 16 light sources in the software renderer. If, by default, both hardware and software renderers are initialized, then the minimum common number of light sources will be used. Thus, both renderers will have eight light sources.

9. 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.

Programming Considerations

There are very few platform-specific issues to take into account when developing AVS modules on SGI workstations.

Compiling and Linking Modules

AVS 5.5 was compiled using the 7.2 versions of the FORTRAN and C compilers and their associated libraries that accompany the IRIX 6.5 system.

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.

Use the Example Makefile as Template

You should use *<installdir>/avs/examples/Makefile* as the template for your own FORTRAN and C module make files. This make file in turn includes the file *<installdir>/avs/include/Makeinclude* that contains additional macro definitions appropriate to the SGI platform, its compilers, compiler options, include files, and libraries.

Compiling with C

As an examination of the *Makeinclude* file would show, when compiling modules written in C you must use the *cc* compiler with the **-cckr** option. This option uses a C language standard as defined in the Kernighan and Ritchie C book, rather than ANSI C. AVS requires the somewhat greater leniency afforded by **-cckr**.

FORTRAN Modules

On 32-bit platforms, pointer and integer data types are the same size and have historically been used as if they were the same. On most 64-bit platforms (Compaq Tru64 UNIX and SGI IRIX 6.5 64-bit (*SGN64*)) pointers are 8 bytes and integers are 4 bytes; the two data types can NOT be used interchangeably. See the section on "FORTRAN Modules on 64-bit systems" in the *UNIX Platforms* chapter for critical information on this topic.

AVS on SGI Workstations: Known Problems

This section lists the known problems with this release of AVS 5.5 that are unique to SGI platforms.

X8328: OpenGL windows in Geometry Viewer cannot follow AVS visual

Problem Description:

The OpenGL **Geometry Viewer** scene windows on the SGI are governed by the GLX interaction and use their own visuals. They cannot follow the LUI_Visual representation of the X server visual that AVS uses.

These problems only occur with the pixmap data type.

If you construct a network that attempts to post-process the pixmap output of the **render geometry** module (e.g., **render geometry**--> **pixmap to image**-->**display image**), the following anomalies occur:

- Your machine is true color, but you have not set the AVS visual to true color, and thus end up using a pseudo color visual. The **pixmap to image** module will produce a solid black image.

- You are in double buffer mode with true color. Only every other frame will actually contain an image. The others will be black.

Workaround: Use the **geometry viewer** module and its image output port for scene content postprocessing, not the unsupported **render geometry** module.

X8371: X terminal support on SGI version isn't automatic

Problem Description:

This happens if the user is running remotely and the local machine is not a distributed OpenGL machine. AVS attempts to start up OpenGL and crashes.

Workaround: Use the **-nohw** option when running remotely in this situation.

X8581: Spaceball on SGI requires special serial port treatment

Problem Description:

Using the spaceball with AVS on the SGI requires that the port the spaceball is connected to be "turned off" in the SGI System Manager utility and then made read/write accessible.

There is an option for serial ports in the SGI OS called "SpaceBall", but setting this option seems to interfere with the spaceball interface. Also, the port (usually */dev/ttyd1*) is only writable by default and is reset to this when altered with the System Manager.

Workaround: Turn off serial port in software (using the System Manager): *chmod a+r /dev/ttyd1* (whatever is appropriate).

X10694: Texture mapping has strange limitations

Problem Description:

The SGI Reality Engine can do 3D texture mapping, but the dimensions of the map must all be powers of 2, and the size must not exceed a certain amount (depending on how much Reality Engine memory you have - typically 128x128x64). Unpredictable results or crashes may result if these are not observed.

X10785: Dual headed operation with different visuals may not work

Problem Description:

If AVS is run on one host and tries to display its **Geometry Viewer** window on a display with a different default visual, problems can occur finding enough colors.

X10301: Module Generator FORTRAN should use integer∗8

Problem Description:

The Module Generator fails to provide special %IFDEF statements for pointer parameters like the example FORTRAN modules do or to directly use INTEGER∗8. Example:

```
      integer function colorizer2(in_field, cmap, out_field)
C %IFDEF ALPHA_OSF
      integer*8 in_field, cmap, out_field
C %ELSE
C      integer in_field, cmap, out_field
C %ENDIF ALPHA_OSF
```

16241: Stereo Graphics displays incorrectly on Octane and Onyx

Problem Description:

AVS stereo mode may not work on some SGI Octane and Onyx platforms. Problems have been reported in which the system tries to go into stereo mode but the display isn't right, showing top/bottom split images, etc.

We were not able to resolve this by release time but did find information suggesting that SGI patches may be available. It has been reported that patches 1822, 1945, and 1953 break quadbuffer stereo on the MXI graphics cards; this is supposedly fixed with patch 2289. It has also been reported that patch 2290 breaks full screen stereo on the Octane; patch 2448 supposedly fixes this problem. For further information check with SGI Customer Support or visit *http://reality.sgi.com/christig/stereo.html*.

Problem: Screen Flicker from Reset Hertz Rate

Problem Description:

On newer SGI platforms, when you exit stereo mode the monitor is automatically reset to 60HZ. This results in screen flicker.

To reset to 72HZ, type:

```
/usr/gfx/setmon  72HZ
```

The man page for *setmon* describes other ways in which you can control the monitor.

Problem: Label color affected by 2D texture mapping

Problem Description:

Use of 2D Texture mapping in a view may affect the color of labels in the same view.

Problem: XDR Reading Short data type

Problem Description:

Some problems have been reported reading the "short" data type in the XDR format.

18326: IRIX 6.5 FlexLM conflict with SGI compiler licenses

Problem Description:

SGI started using FlexLM to license its compilers in IRIX 6.5. It stores the licenses in the */var/flexlm* directory by default. The */var/flexlm/licensefile.db* file usually contains a list of the files to check for licenses and this usually includes the default path AVS normally uses, */usr/local/flexlm/licenses/license.dat*.

This will normally all work without problems but if the end user redefines the LM_LICENSE_FILE environment variable to identify a demo license or other local license, the SGI C compilers won't be able to find their licenses and will fail.

Workaround:

- don't use LM_LICENSE_FILE if possible
- combine the /var/flexlm license into the new license file you are working with
- set LM_LICENSE_FILE to a longer list of multiple files to check including the /var/flexlm license that has the compiler licenses.

AVS on SGI Workstations: Fixed Problems

This is the list of fixed problems for AVS 5.5 on the SGI platforms.

7823: Coroutine integer output does not work on SG3 (N64)

Problem Description:

The animated integer module was not working on the N64 platform in AVS 5.4 - modules downstream only received "0" because the wrong half of the 64 bit integer was being communicated. This has been fixed.

13157: Memory leak in SGI texture mapping

Problem Description:

Using texture mapping in the Geometry Viewer on the SGI platforms would leak when the textures were changed as the low level copy of the texture map was not being freed. This has been fixed by reenabling shared OpenGL contexts within the renderer, made possible by improvements in the IRIX 6.5 release.

17860: SGI stereo initialization problems on Infinite Reality

Problem Description:

Problems were reported initializing stereo on the IR graphics adapters; attempts to enter stereo mode, left all-red view windows and prevented exiting stereo using the control-mouse-left binding. A temporary workaround of using the Geometry Viewer/Camera/Stereo button three times (on/off/on) seemed to initialize stereo correctly. This has been fixed.