WoS

Semper 6 *Plus*

NOTES FOR SILICON GRAPHICS INSTALLATIONS

Synoptics

Trademarks:

Semper is a registered trademark of *Synoptics Limited*
IRIX is a trademark of Silicon Graphics, Inc.

Manual number: SEM032
Revision date: 22/3/91

Synoptics Limited, 271 Cambridge Science Park, Milton Road, Cambridge, CB4 4WE, U.K. Tel: (0223) 423223, Fax: (0223) 420020, Int. Tel: +44 223 423223, Int. Fax: +44 223 420020.

Synoptics Limited, Paragon Towers, 233 Needham Street, Newton, MA 02164, USA. Tel: 617 527 4461, Fax: 617 527 4084.

Table OF CONTENTS

Installing Semper 6 *Plus* on a Silicon Graphics

Upgrade Notes

Installation Specific Commands

Installing

SEMPER 6 *PLUS* ON A SILICON GRAPHICS

Overview

Follow the steps given below to install Semper 6 *Plus* on your Silicon Graphics workstation.

Step 1

Before installing Semper 6 *Plus* you will require a suitable account on the target Silicon Graphics. The usual procedure is to create a user called *semper* if no suitable account exists. The account requires no special privileges, but should be configured to run the C shell (/bin/csh).

Approximately 10 megabytes of disc space are required to install Semper 6 *Plus*.

Step 2

Login as user *semper* (or under the username you created). In a suitable directory, which is the home directory for user *semper*, retrieve the contents of the distribution archive tape as follows:

Mount the tape in the drive, then copy Semper files from tape by typing the following commands:

```
dd   if=/dev/tape conv=swab | tar oxvf -
```

Step 3

Semper should now be on your system. To enter Semper, change directory to that containing the Semper executable, or ensure that the directory is on your PATH (you will already be in the correct directory if you created a user *semper*). Type the command:

```
semper
```

and the system will start up and run through a brief introduction.

Upgrade

NOTES

Overview

This document describes the changes made to Semper on a Silicon Graphics workstation at the 6.3 upgrade, designed to run under version 3.3.1 of the IRIX operating system. This latest version of Semper brings with it a number of new features and corrects some outstanding problems. It supports the following features in addition to the standard image processing facilities:

1. **Window based display**

   - False colour (all systems) or full colour (systems with 24 bit-planes)
   - 8 overlays with configurable colour and visibility
   - Separate graphics, cursor and rubberband overlays
   - New **overlay** command
   - Image look-up tables
   - Support for full colour look-up tables (systems with 24 bit-planes)
   - Display window of variable size and position
   - Separate menu interface window
   - Proper window management and repair

2. **Configurable disc cache**

   - Any number of cache buffers
   - Cache buffer size a multiple of page size (4096 bytes)
   - Default of 32 buffers of 8192 bytes (256K disc cache)
   - New **cache** command

3. **Interactive input**

   - Support for tracking mouse movements and button presses
   - Function and cursor keys supported
   - Command line editing
   - Emulation of Unix terminal control codes (**intr**, **kill**, **erase** and **rprnt**)
   - Additional <break> key

4. **Batch mode – standard input from file instead of the terminal**

5. **Access to images stored in Silicon Graphics' image file format**

   - Compressed and uncompressed image formats
   - New **iget** command
   - New **iput** command

These features are described in detail in the following sections:

1. Window based display

The display window is now fully buffered and is repainted whenever it is necessary.

The resolution of the display window depends on the number of bit-planes supported by the hardware. Some graphics hardware supports 12 and 24 bit-planes. For these systems, false colour images with 256 levels or full colour images with 256 levels for red, green and blue are supported. Look-up tables for full colour images are now also supported. For systems with only 8 bit-planes, only false colour images with less than 256 levels are supported.

By default, a false colour display window is opened when you use the **assign** command. A full colour window is obtained with the following command:

```
assign display frames 3
```

Remember that in order to display full colour images, you also have to create a partition with the requisite number of frames, for example:

```
partition 1 size 300 top left frames 1,3
```

A total of 8 overlays are now supported on all systems. With the **overlay** command you can control overlay visibility as well as colour. The overlays are numbered from 1 to 8 and where overlay data overlaps, the highest numbered overlay is displayed. Cursor and rubberbanding functions are carried out using one overlay each. Different overlays must be used for displaying graphics, cursor and rubberbanding.

Changes of overlay visibility and of the image look-up table for a full colour display requires that the display window be repainted.

Systems with only 8 bit-planes can display a total of 256 colours. Of these, 8 colours are set aside for the overlays and a further 44 colours are left for the window manager, leaving a total of 204 levels for displaying images. The colour indices set aside for the window manager are 0 to 15, 32 to 56, 196, 251 and 255. Colour 196 corresponds to the standard pale blue background colour and colour 251 is used for displaying the tool-chest menu icons.

You can use the **size** and **position** keys with the **assign display** command to control the size and position of the display window on the screen. If only the *x* dimension is specified, the *y* dimension defaults to the *x* dimension. If no dimensions are specified, a default size of 768 by 512 is used. The window position is defined as the offset from the bottom left of the screen to the bottom left of the display window. If the window position is not specified, you have to position the window yourself on the screen using the mouse. Once the window has been opened it can be resized to any size up to the initial size and also repositioned. The initial size of the window determines the size of the display buffer and all the data in the buffer is accessible from Semper at all times regardless of the current size, position or visibility of the window.

2. Configurable disc cache

Semper's disc caching scheme has been very much improved. Instead of a fixed size cache of 8 buffers of 8192 bytes each, you can now specify the number and size of cache buffers at any time by using the new **cache** command. The disc cache starts off with 32 buffers of 8192 bytes.

A more efficient caching algorithm has been used. This, in combination with the larger initial size of the disc cache, should improve the speed of access to data for most image processing commands. If, however, the cache is large enough for there to be no need to access the hard disc, you should see a further increase in performance. For example, a 512 by 512 FFT which takes about 30 seconds to process on a 4D/20 Personal Iris, will run in 20 seconds when the data is fully cached.

Comparisons of timings with and without Semper's disc cache are sometimes difficult to explain because underlying all access to the hard disc is the Unix disc cache, which Semper cannot bypass when it has to access data on the hard disc. The performance of the Unix disc cache can be very unpredictable because it will be affected by any other processes which access the disc while Semper is running. Having said that, you will always get better and more consistent performance when data is fully cached in Semper's own disc cache.

When increasing the size of the disc cache, you must be very careful not to use all of the free memory. When the amount of free memory comes down to about 500 Kbytes, the operating system will start to page out memory more frequently and the performance of the whole machine will suffer. You can monitor memory use by running the system monitor program using the Unix command:

```
gr_osview -a
```

3. Interactive input

Semper can track mouse movements anywhere on the screen. When mouse movements are fed back onto the display window, for example when using the **xwires** command, the mouse position is clipped to fall inside the display window. This makes the cursor in the display window track round the edge of the window as the screen cursor is moved outside the display window.

Mouse button presses and any keyboard input are monitored when any of Semper's windows has input focus. The use of the function keys and the cursor keys is supported (see the command **inkey**).

The Unix terminal configuration defines keycodes for break (intr), character delete (erase), line delete (kill) and line refresh (rprnt). These can be examined or changed with the Unix command **stty**. In addition to these, the Pause, Insert, Home and End keys and the control codes ^B and ^E have special meanings:

<intr> or **Pause**	Abandon request
<erase>	Backspace and delete character
<kill>	Delete line
<rprnt>	Refresh line
Home or ^B	Start of line
End or ^E	End of line
Insert	Insert/replace mode

The keycodes returned for the function keys, cursor keys and the special keys listed overleaf (except abandon requests) are described in *Appendix J* of the following manual:

Semper 6 Command Reference Manual

4. Batch mode

If Semper's standard input stream is not connected to a terminal device, Semper will run in batch mode. This causes Semper to set the *batch* variable to 1 instead of zero when it starts up. All forms of interactive input are disabled and any attempt to use a command that requires interactive input will be faulted.

5. Silicon Graphics image file format

The new commands **iput** and **iget** provide access to Silicon Graphics' image file format. Images stored in this form can be manipulated by native commands like **icut, ipaste, snapshot** and **istat.** The file format includes the facility to compress the image data (run-length encoding). **iget** can read in compressed image files and **iput** will always write out images in compressed form.

Warnings

The following caveats are issued with the Semper 6.3 upgrade:

Memory requirements

Semper can be run without difficulty on systems with the basic 8 Mbytes of memory. Tests indicate, however, that this is not enough to support the efficient operation of large display windows or a large disc cache (a full screen, full colour display window requires 5 Mbytes of memory, and a disc cache of 3 to 4 Mbytes for processing large images is not unreasonable). We would recommend a minimum of 12 Mbytes of memory for running Semper and 16 Mbytes or more of memory for efficient processing of large images.

Fortran compiler optimization

A bug in the global optimizer used by the Fortran compiler has meant that all of Semper's Fortran code has had to be compiled with a lower level of optimization. This has reduced the speed of execution for most of Semper by a factor of 40 to 50 percent. The problem has been reported to *Silicon Graphics, Inc.* If a solution is provided by them, it will be incorporated into the next release of Semper after the solution becomes available.

Installation

SPECIFIC COMMANDS

This section describes the commands that are specific to Semper systems running on a Silicon Graphics workstation at the Semper 6.3 upgrade. The following commands are documented:

- **cache**
- **iget**
- **input**
- **iput**
- **output**
- **overlay**

If you have a copy of the *Semper 6 Command Reference* you may like to replace the existing command descriptions with the descriptions given in the following pages.

For an overview of the special facilities provided on a Silicon Graphics workstation, type the following command at the Semper prompt:

```
help 4sight
```

cache

This syntax is specific to...
Silicon Graphics workstations

keys:	**number**	*<number>*	redefine number of cache buffers
	size	*<number>*	redefine size of cache buffer (in bytes)
options:	**flush**		flush contents of cache buffer to disc
	show		list current settings for cache buffer parameters

Use the **cache** command to manage the way in which input/output to disc is cached in virtual memory.

Examples

```
cache show
```

This command lists the current disc cache parameter settings – cache buffer number and size and total cache size (and operating system page size, if relevant).

```
cache number 200
```

This command changes the number of cache buffers to 200, keeping the buffer size the same.

```
cache number 20 size 65536
```

This command reconfigures the cache to consist of 20 buffers of 65536 bytes each (total cache size = 1310720 bytes).

```
cache flush
```

This command causes all data modified in the cache to be written to disc.

Description

Disc data is buffered in virtual memory before it is accessed by Semper processing commands. This can lead to significant reductions in data traffic to and from the hard disc, especially when all the data for a particular image processing operation is cached.

The disc cache consists of a number of buffers of a certain size. The buffer size must be a multiple of 8 and, on some systems, it may also need to be a multiple of the operating system's page size. Use the **cache** command to alter the number and size of the cache buffers.

If the number of buffers or the buffer size is zero, disc data will be accessed directly from the hard disc. The disc cache helps to reduce the number of input/output requests. Accessing the hard disc without a cache will tend to give lower performance.

cache

The size of the cache needs to be carefully set so that it is not so large that it puts too heavy a burden on the operating system. If the size of the cache exceeds the amount of free memory, data will be swapped out of the memory to make room. If this leads to the point where active processes have to be swapped out, the performance of the whole system will suffer.

Reconfiguring the disc cache causes its contents to be written to disc, as if the **flush** option had been specified. If the disc cache is large, this could take a noticeable length of time. Likewise, when a disc device is deassigned, its contents are flushed out to disc. The **flush** command has the same effect as using the **flush** option.

The **show** option causes the current disc cache parameters to be listed on the console output stream. You should ignore the information about the cache size given by the **show system** command as this is no longer relevant.

Notes

see also: **assign, deassign, flush**

Defaults and Ranges

keys/options	defaults	range
number	*none*	positive integers
size	*none*	positive integers

iget

This syntax is specific to:
Silicon Graphics workstations

keys:	[to]	*<number>*	specify number of output picture
	name	*'<text>'*	specify file from which to read picture

Use the iget command to read in an image stored in a file using the Silicon Graphics' image file format.

Examples

```
iget dis:1 name 'cat.rgb'
```

This command transfers the image stored in the file *cat.rgb* directly to display partition 1. Note that Unix file names are case sensitive.

Description

The **iget** command is designed to read an image stored in Silicon Graphics file format into Semper. The **iget** command can read compressed (run-length encoded) images. Note that before you start Semper, you can check the file header information for image files using the Unix command **istat** (see the corresponding Unix manual entry for details of this command).

Notes

see also: **iput**

Defaults and Ranges

keys/options	defaults	range
[to]	current picture, held in the variable *select*	valid picture number
name	*none*; prompts if interactive	valid file name

input

This syntax is specific to...
Silicon Graphics workstations

keys:	[to]	*<number>*	picture to be read
	name	*'<text>'*	name of Semper 6 file containing picture data
	size	*<x>,<y>,<z>*	if ***raw***, specify the size of a raw binary data image
	skip	*<number>*	if ***raw***, specify the number of bytes to skip in the file before reading data
	map	*<number>*	picture containing Sun Raster colormap data
options:	**raw**		read a raw binary data file
	raster		read a Sun Raster file
	byte/integer/fp/complex		if ***raw***, read data in byte (8 bit), integer (16 bit), floating point or complex form
	swap		if ***raw***, read data in *Motorola* byte order

Use **input** to read pictures from files created by Semper 6 on workstations or PCs (using the **output** command) or by other selected programs.

Examples

```
input 5 name 'newdata'
```

This command reads picture 5 from the Semper 6 file called *newdata.pic*.

```
input 125 raw name 'binary' size 512,512
```

This command reads picture 125, of size 512 by 512 pixels, from the raw binary file called *binary.bin*.

```
input 33 raw integer name 'mydata' size 640,480 swap
```

This command reads picture 33, of size 640 by 480 pixels, as 16 bit integers with *Motorola* byte ordering from the file *mydata.bin*.

```
input 1 raw name 'special.pic' size 128,128 skip 64
```

This command reads picture 1, of size 128 by 128 pixels, from the raw binary file *special.pic*, skipping the first 64 bytes of the file.

input

```
input 1 raster name 'dump.rff' map 2
```

This command reads a Sun raster file, called *dump.rff*, into picture 1, storing any colormap information in picture 2.

Description

By default, **input** expects to read a Semper 6 data file unless you specify the **raw** or **raster** options. The default extension for Semper 6 files is *.pic*.

If you specify the **raw** option, Semper reads a raw binary file. The default extension for raw binary files is *.bin*. If you specify the **raw** option, you also need to specify the size of the file using the **size** key, with bytes starting with the top left pixel of layer 1, reading along the row. You can also use the **skip** key with the **raw** option to specify the number of bytes to skip before reading data from a file. This key is useful if you have picture files that contain header information of a known size.

The bytes are ordered in the Semper 6 data file so that the least significant byte appears at the first (lowest) address (*Intel* format). You can use the **swap** option with **raw** to specify that the binary data uses the *Motorola* packing format. You can also specify the form of data input using the options **byte**, **integer**, **fp** and **complex** with the **raw** option.

If you specify the **raster** option, Semper reads a Sun Raster image file. The default extension for raster files is *.rff*. Currently only files with 8 bits per pixel are accepted. The file packing types support the old and new unpacked types 0 (RT_OLD) and 1 (RT_STANDARD), and also the byte encoded type 2 (RT_BYTE_ENCODED). If you use the **map** key with the **raster** option, Semper stores any colormap data that is present in the specified picture number. This information can be used to remap the image data that you read in using the Semper **map** command. If the map data has three rows (RGB colormap) you can remap the original image as follows:

```
input 1 raster name 'myfile.rff' map 2      ;! read the data
project 2 vertically fp                     ;! sum columns
calculate :2/3 to :3                        ;! average
copy 3 2 byte                               ;! restore as byte map
map 1 to 3 with 2                           ;! and map the data
```

Files written using the **output** command on a PC can be read into a UNIX workstation linked over a *PC-NFS* type network. Note that **input** searches for files in the current directory and then throughout the DOS PATH. You can avoid a time-consuming path scan by specifying a full path name with the filename.

Notes

see also: **output, read, write**

input

Defaults and Ranges

keys/options	defaults	range
[tc]	current picture, held in the variable *select*	valid picture number
name	*none*; you are prompted for a name	valid filename
size	*none*	valid picture size
skip	*none*	positive integers
map	*none*	valid picture number

iput

This syntax is specific to:
Silicon Graphics workstations

keys:	**[from]**	*<number>*	specify number of source picture
	name	*'<text>'*	write picture to the specified file
options:	**new**		overwrite an existing output file

Use the **iput** command to write a Semper image to file using the Silicon Graphics' file format.

Examples

```
iput 3:2 name 'picture.img'
```

This command writes picture 3:2 out to an image file called *picture.img*. Note that Unix file names are case sensitive.

```
iput 1 name 'mona.img' new
```

This command writes picture 1 to an existing image file called *mona.img*, overwriting the original contents of the file.

Description

The **iput** command is designed to write a Semper image to file in Silicon Graphics' image file format. The file can then be read by other facilities on the Silicon Graphics, such as **istat**, **icut**, **ipaste** and **snapshot** (consult the corresponding Unix manual entries for these commands). Note that the data written to file is run-length encoded.

Use the **name** key to specify an image file name and the **new** option to overwrite the contents of an existing image file.

Notes

see also: **iget**

Defaults and Ranges

keys/options	defaults	range
[from]	current picture, held in the variable *select*	valid picture number
name	*none*; prompts if interactive	valid file name

output

This syntax is specific to...
Silicon Graphics workstations

keys:	[from]	*<number>*	source picture
	name	*'<text>'*	name of file to contain output data
options:	**raw**		write a raw binary data file
	raster		write a Sun Raster format image
	new		overwrite an output file if it already exists
	unlabelled		write without including picture label information
	swap		if *raw*, write data in *Motorola* byte order
	byte/integer/fp/complex		if *raw*, write data in byte (8 bit), integer (16 bit), floating point or complex form

Use **output** to output pictures in one of a number of fast binary forms.

Examples

```
output 23 name 'binary' unlabelled
```

This command writes picture 23 to the file *binary.pic* excluding the picture label.

```
output 12 raw name 'rawdata'
```

This command writes picture 12 to the file *rawdata.bin* as a byte stream.

```
output 12 raw byte name 'rawdata'
```

This command writes picture 12 to the file *rawdata.bin* as a byte stream.

```
output 14 raw name 'local.dmp' swap
```

This command writes picture 14 to the file *local.dmp* using *Motorola* byte ordering.

```
output 303 raster name 'screen'
```

This command writes picture 303 as a Sun Raster file named *screen.rff*.

output

Description

Use the **output** command to write a Semper 6 data file, unless you specify the **raw** or **raster** options. Note that **output** will only overwrite an existing file of the same name if you specify the **new** option.

If you specify the **raw** option, Semper produces a raw binary file. A raw data file is a byte stream starting with the top left pixel of layer 1, outputting along each row. Pixels are written to a raw data file using the form of the output picture, unless you specify one of the data form options **byte**, **integer**, **fp** or **complex.**

By default, the bytes are ordered in the Semper 6 data file so that the least significant byte appears at the first (lowest) address (*Intel* format). You can also write raw data files using *Motorola* byte ordering, if you specify the **swap** option in conjunction with **raw**.

If you specify the **raster** option, Semper produces a Sun Raster image. Currently these are always written with a depth of 8 bits per pixel, using standard packing (type 1 – RT_STANDARD) and with no colormap information (RMT_NONE).

With the exception of **raw** binary files, Semper includes the picture label unless you specify the **unlabelled** option. **output** stores the same data in the file as the **write** command.

The default extension for Semper 6 files is *.pic*, for raw binary files *.bin* and for raster files *.fff.*

Files written using the **output** command on a PC can be read into a UNIX workstation, linked over a *PC-NFS* type network. You can recreate pictures that are produced by **output**, using the Semper command **input.**

Notes

see also: **input, read, write**

Defaults and Ranges

keys/options	defaults	range
[from]	current picture, held in the variable *select*	valid picture number
name	*none*; if interactive, you are prompted for a name	valid file name

overlay

This syntax is specific to:
Silicon Graphics workstations

keys:	[*number*]		specify overlay number
	rgb	*<n1>,<n2>,<n3>*	specify colour values in percentage terms
	hsv	*<n1>,<n2>,<n3>*	specify hue, saturation and brightness
options:	**black/red/green/blue/cyan**		specify colour for overlay plane
	magenta/yellow/white		
	on/off		make the overlay plane visible or invisible
	graphics/rubberband/cursor		specify function of overlay
	show		list current overlay settings

You use the **overlay** command to control the colour and visibility of the display window's eight overlays. Display annotation or graphics is directed to the current graphics overlay, rubberband lines are displayed in the rubberband overlay and the cursor is displayed in the cursor overlay.

Examples

```
overlay show
```

This command lists the current overlay settings.

```
overlay cyan
```

This command sets the colour of overlay 1 to cyan.

```
overlay 3 hsv 60,50,100 cursor
```

This command sets the colour of overlay 3 to a desaturated yellow and makes this the cursor overlay.

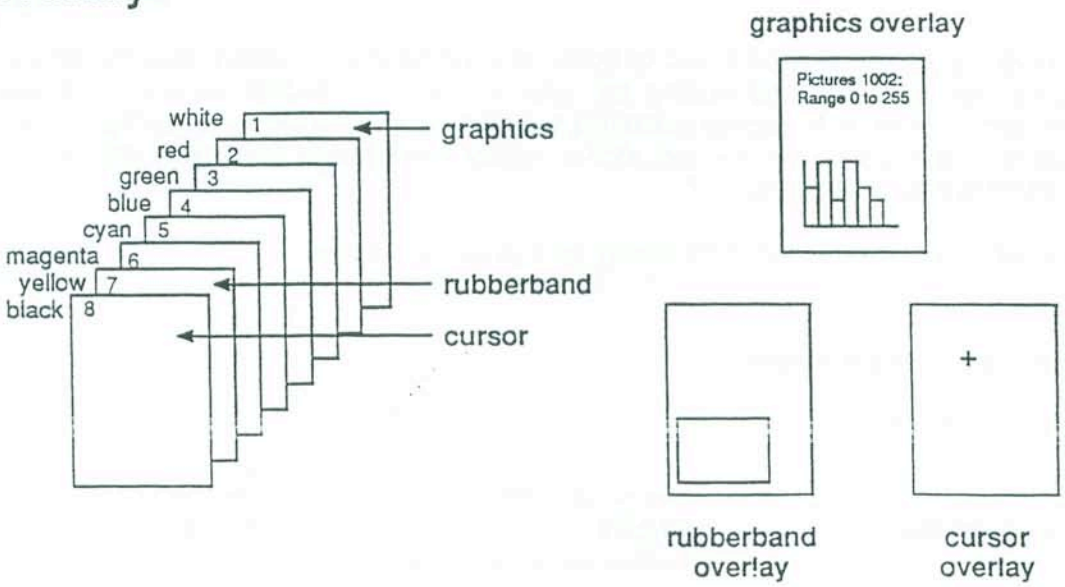
```
overlay 2 off
```

This command turns off overlay 2.

Description

The display window on a Silicon Graphics workstation has eight overlay bit-planes numbered from 1 to 8. When the display window is opened, overlays 1, 7 and 8 are selected as the graphics, rubberband and cursor overlays, all overlays are turned on and cleared and the colours for overlays 1 to 8 are set to white, red, green, blue, cyan, magenta, yellow and red respectively. A primary use of the **overlay** command is to specify the colour of the annotation which is held in a graphics overlay. The diagram overleaf illustrates the default overlay setting.

overlay



The **overlay** command changes only the overlay settings that are specified with the command. The remaining overlay settings are left unchanged. Use the **number** key to specify the overlay number. Note that the **number** key always defaults to 1.

You can specify a colour for an overlay plane in one of three ways:

- **rgb** key
- **hsv** key
- **black/red/green/blue/cyan/magenta/yellow/white** colour options

If you need to specify the colour exactly, you can specify the **rgb** components of the colour as percentages with the **rgb** key, or the hue, saturation and brightness of the colour with the **hsv** key. The saturation and brightness values must also be specified as percentages. If the colour you want is one of the eight primary or secondary colours, you can use the corresponding colour option as a convenient alternative. Note that you may not specify more than one of the options **black, white, red, green, blue, cyan, magenta** and **yellow** and the **rgb** and **hsv** keys at one time.

To turn an overlay on or off (to make an overlay visible or invisible) you can specify the **on** or **off** options. Note that by turning an overlay on or off you do not affect the data stored in the overlay bit-planes.

You may use one of the **graphics, rubberband** or **cursor** options to make the specified overlay the current graphics, rubberband or cursor overlay. You are not allowed to specify more than one function for a given overlay. For example, the following would not be allowed:

```
overlay 2 graphics; overlay 2 rubberband
```

All display annotation is directed into the current graphics overlay. Note that the **erase overlay** command erases only the current graphics overlay and leaves the other seven overlays unchanged. To clear all the overlays, type the following command line:

```
for n=1,8
overlay n graphics; erase frame overlay;
loop n
```

overlay

The cursor and any rubberband lines or boxes (see the **xwires** and **pdraw** commands) are displayed by overwriting and erasing the data in the corresponding overlays. You are recommended to retain the highest numbered overlay for displaying the cursor so that it is not obscured by any of the other overlays. Where overlay information is overlapped, the highest numbered overlay is displayed.

The **show** option lists all the current overlay settings on the console.

Notes

see also: **4sight, xwires**

Defaults and Ranges

keys/options	defaults	range
[number]	1	1 to number of overlays
rgb	*none*	real number expressing a percentage
hsv	*none*	hue = integer 0 to 360 saturation = percentage brightness = percentage