(void)enterSVGAMode

Implemented by subclasses to put the display into SVGA mode. This method is invoked by the sys
such as when the window server starts running. This method should set up all the registers necessa
mode, set the color palette, and clear the screen.

You should set the color palette to contain values for the four supported shades of gray in the first
the entries should be zeroed out. NeXT drivers currently use the palette values shown in the follow

revertToVGAMode

Implements this method, as described in the IOScreenEvents protocol specification. You should ne
implement this method.

initFromDeviceDescription:deviceDescription

Invokes initFromDeviceDescription: on super. If successful, sets the unit number and the name (to
followed by the unit number). Frees itself if initialization was unsuccessful.

Subclasses must implement this method so that it performs all initialization necessary to set up the
After invoking initFromDeviceDescription: on super, this method should determine its mode (invo
count: or selectMode:count:valid:, if necessary) and set [self displayMode] to the IODisplayInfo ap
The driver should finish by invoking mapFrameBufferAtPhysicalAddress:length: and setting the IO
frameBuffer field to the value returned.

If possible, this method should check the hardware to see if it matches the IOConfigTable. If the ha
the driver should do what it can to ensure that the display is still usable.

(vm_address_t)mapFrameBufferAtPhysicalAddress:(unsigned int)address length:(int)numByte

Maps the physical memory for this instance into virtual memory for use by the device driver. If ad
maps the physical memory corresponding to local memory range 0, and numBytes is ignored. If ad
reserved resources are overriddenÐ address is used as the physical memory address and numBytes
The mapped memory range is cached as IO_WriteThrough.

moveCursor:(Point *)cursorLoc
frame:(int)frame
token:(int)token

Implements this method, as described by the IOScreenEvents protocol. You should never need to i
this method.

(void)restorePlaneAndSegmentSettings

Implemented by subclasses to restore the plane and segment settings to the saved values. This meth
IOSVGADisplay's cursor handling methods. The cursor handling methods invoke savePlaneAndS
whatever is necessary to update the cursor, and then invoke restorePlaneAndSegmentSettings to re
state.

Here's an example of implementing this method by saving the current settings into subclass-define

be used as a standard VGA device. Implementing this method usually consists of setting registers t
VGA.

(void)savePlaneAndSegmentSettings

Implemented by subclasses to save the current plane and segment settings. This method is invoked
cursor handling methods. The cursor handling methods invoke savePlaneAndSegmentSettings, do
to update the cursor, and then invoke restorePlaneAndSegmentSettings to restore the display's stat

Each invocation of savePlaneAndSegmentSettings is followed by exactly one invocation of
restorePlaneAndSegmentSettings, with no intervening invocations of savePlaneAndSegmentSettin
driver only has to remember one group of settings at a time.

Here's an example of implementing this method by saving the current settings into subclass-define

(int)selectMode:(const IODisplayInfo *)modeList count:(int)count

Invokes selectMode:count:valid:, specifying 0 for the last argument.

(int)selectMode:(const IODisplayInfo *)modeList
count:(int)count
valid:(const BOOL *)isValid

Determines which IODisplayInfo in the driver-supplied modeList matches the value of the ªDispla
device's IOConfigTable. Drivers that support multiple advanced modes should invoke this method
When the driver receives a enterSVGAMode message, it should enter the mode selected by this m
doesn't find a valid mode, the driver should determine a mode that will work.

The ªDisplay Modeº key is a configuration key that can be used by drivers to support multiple mo
Hz and 72 Hz. IODisplayInfo is defined in the header file driverkit/displayDefs.h.

The modeList argument should contain an IODisplayInfo for each advanced mode the driver suppo
argument should specify the number of IODisplayInfos in modeList. isValid should either be 0 (in
ignored) or an array that corresponds to the modeList. If isValid[1] is NO, for example, then this m
IODisplayInfo pointed to by modeList[1].

If this method finds a match, it returns the index of the matching IODisplayInfo in modeList. If the
is missing or its value is improperly formatted, or if a corresponding IODisplayInfo isn't found, thi

See the IODisplay class description for information on display modes and the IODisplayInfo type.

setBrightness:(int)level token:(int)token

(IOReturn)setIntValues:(unsigned int *)parameterArray
forParameter:(IOParameterName)parameterName
count:(unsigned int)count

Handles NeXT-internal parameters specific to IOSVGADisplays forwards the handling of all other

setIntValues:forParameter:count: (IODevice)

(void)setReadPlane:(unsigned char)planeNum

Implemented by subclasses to set which of two planes the display subsystem will read from. Only
at a time. Here's an example of implementing this method.

setWritePlane:

(void)setReadSegment:(unsigned char)segmentNum

Implemented by the subclass to set the 64KB segment the display subsystem will read from.

setReadPlane:

(void)setWriteSegment:(unsigned char)segmentNum

Implemented by the subclass to set the 64KB segment the display subsystem will read from.

setReadSegment:

showCursor:(Point *)cursorLoc
frame:(int)frame
token:(int)token

Implements this method, as described in the IOScreenEvents protocol specification. You should ne
implement this method.