NAME
   vidcontrol - system video console control and configuration utility

SYNOPSIS
   vidcontrol [**-CdHLPpx**] [**-b** *color*] [**-c** *appearance*] [**-E** *emulator*] [**-f** [[*size*] *file*]] [**-g** *geometry*] [**-h** *size*]
           [**-i active** | **adapter** | **mode**] [**-l** *screen_map*] [**-M** *char*] [**-m on** | **off**] [**-r** *foreground background*]
           [**-S on** | **off**] [**-s** *number*] [**-T xterm** | **cons25**] [**-t** *N* | **off**] [*mode*] [*foreground* [*background*]]
           [**show**]

DESCRIPTION
   The **vidcontrol** utility is used to set various options for the syscons(4) or vt(4) console driver, such as
   video mode, colors, cursor shape, screen output map, font, and screen saver timeout. Only a small
   subset of options is supported by vt(4). Unsupported options lead to error messages, typically including
   the text "Inappropriate ioctl for device".

   The following command line options are supported:

   mode   Select a new video mode. The modes currently recognized are: *80x25*, *80x30*, *80x43*, *80x50*,
           80x60, *132x25*, *132x30*, *132x43*, *132x50*, *132x60*, *VGA_40x25*, *VGA_80x25*, *VGA_80x30*,
           VGA_80x50, *VGA_80x60*, *VGA_90x25*, *VGA_90x30*, *VGA_90x43*, *VGA_90x50*,
           VGA_90x60, *EGA_80x25*, *EGA_80x43*, *VESA_132x25*, *VESA_132x43*, *VESA_132x50*,
           VESA_132x60. The raster text mode *VESA_800x600* can also be chosen. Alternatively, a
           mode can be specified with its number by using a mode name of the form MODE_*<NUMBER>*.
           A list of valid mode numbers can be obtained with the **-i mode** option. See *Video Mode Support*
           below.

   foreground [*background*]
           Change colors when displaying text. Specify the foreground color (e.g., "vidcontrol white"), or
           both a foreground and background colors (e.g., "vidcontrol yellow blue"). Use the **show**
           command below to see available colors.

   show   See the supported colors on a given platform.

   -b *color*
           Set border color to *color*. This option may not be always supported by the video driver.

   -C   Clear the history buffer.

   -c *setting*[,*setting ...*]
           Change the cursor appearance. The change is specified by a non-empty comma-separated list of

*setting*s. Each *setting* overrides or modifies previous ones in left to right order.

The following override *setting*s are available:

normal

Set to a block covering 1 character cell, with a configuration-dependent coloring that should be at worst inverse video.

destructive

Set to a blinking sub-block with **height** scanlines starting at **base**. The name "destructive" is bad for backwards compatibility. This *setting* should not force destructiveness, and it now only gives destructiveness in some configurations (typically for hardware cursors in text mode). Blinking limits destructiveness. This *setting* should now be spelled **normal**,**blink**,**noblock**. A non-blinking destructive cursor would be unusable, so old versions of **vidcontrol** did not support it, and this version does not have an override for it.

base=*value*, **height**=*value*

Set the specified scanline parameters. These parameters are only active in **noblock** mode. **value** is an integer in any base supported by strtol(3). Setting **height** to 0 turns off the cursor in **noblock** mode. Negative *value*s are silently ignored. Positive *value*s are clamped to fit in the character cell when the cursor is drawn.

The following modifier *setting*s are available:

blink, **noblink**

Set or clear the blinking attribute. This is not quite backwards compatible. In old versions of **vidcontrol**, **blink** was an override to a blinking block.

block, **noblock**

Set or clear the **block** attribute. This attribute is the inverse of the flag CONS_CHAR_CURSOR in the implementation. It deactivates the scanline parameters, and expresses a preference for using a simpler method of implementation. Its inverse does the opposite. When the scanline parameters give a full block, this attribute reduces to a method selection bit. The **block** method tends to give better coloring.

hidden, **nohidden**

Set or clear the hidden attribute.

The following (non-sticky) flags control application of the *setting*s:

charcolors

Apply **base** and **height** to the (character) cursor's list of preferred colors instead of its shape. Beware that the color numbers are raw VGA palette indexes, not ANSI color numbers. The indexes are reduced mod 8, 16 or 256, or ignored, depending on the video mode and renderer.

mousecolors

Colors for the mouse cursor in graphics mode. Like **charcolors**, except there is no preference or sequence; **base** gives the mouse border color and **height** gives the mouse interior color. Together with **charcolors**, this gives 2 selection bits which select between only 3 of 4 sub-destinations of the 4 destinations selected by **default** and **local** (by ignoring **mousecolors** if **charcolors** is also set).

default

Apply the changes to the default settings and then to the active settings, instead of only to the active settings. Together with **local**, this gives 2 selection bits which select between 4 destinations.

shapeonly

Ignore any changes to the **block** and **hidden** attributes.

local Apply the changes to the current vty. The default is to apply them to a global place and copy from there to all vtys.

reset Reset everything. The default is to not reset. When the **local** parameter is specified, the current local settings are reset to default local settings. Otherwise, the current global settings are reset to default global settings and then copied to the current and default settings for all vtys.

show Show the current changes.

-d Print out current output screen map.

-E *emulator*
Set the terminal emulator to *emulator*.

-e Show the active and available terminal emulators.

-f [[*size*] *file*]
Load font *file* for *size* (currently, only **8x8**, **8x14** or **8x16**). The font file can be either uuencoded

or in raw binary format. You can also use the menu-driven vidfont(1) command to load the font of your choice.

Size may be omitted, in this case **vidcontrol** will try to guess it from the size of font file.

When using vt(4) both *size* and *file* can be omitted, and the default font will be loaded.

Note that older video cards, such as MDA and CGA, do not support software font. See also *Video Mode Support* and *EXAMPLES* below and the man page for either syscons(4) or vt(4) (depending on which driver you use).

-g *geometry*

Set the *geometry* of the text mode for the modes with selectable geometry. Currently only raster modes, such as *VESA_800x600*, support this option. See also *Video Mode Support* and *EXAMPLES* below.

-h *size*

Set the size of the history (scrollback) buffer to *size* lines.

-i active

Shows the active vty number.

-i adapter

Shows info about the current video adapter.

-i mode

Shows the possible video modes with the current video hardware.

-l *screen_map*

Install screen output map file from *screen_map*. See also syscons(4) or vt(4) (depending on which driver you use).

-L   Install default screen output map.

-M *char*

Sets the base character used to render the mouse pointer to *char*.

-m on | **off**

Switch the mouse pointer **on** or **off**. Used together with the moused(8) daemon for text mode cut & paste functionality.

-p   Capture the current contents of the video buffer corresponding to the terminal device referred to by standard input. The **vidcontrol** utility writes contents of the video buffer to the standard output in a raw binary format. For details about that format see *Format of Video Buffer Dump* below. Supported only with syscons(4).

-P   Same as **-p**, but dump contents of the video buffer in a plain text format ignoring nonprintable characters and information about text attributes. Supported only with syscons(4).

-H   When used with **-p** or **-P**, it instructs **vidcontrol** to dump full history buffer instead of visible portion of the video buffer only.

-r *foreground background*
           Change reverse mode colors to *foreground* and *background*.

-S on | **off**
           Turn vty switching on or off. When vty switching is off, attempts to switch to a different virtual terminal will fail. (The default is to permit vty switching.) This protection can be easily bypassed when the kernel is compiled with the DDB option. However, you probably should not compile the kernel debugger on a box which is supposed to be physically secure.

-s *number*
           Set the active vty to *number*.

-T xterm | **cons25**
           Switch between xterm and cons25 style terminal emulation.

-t *N* | **off**
           Set the screensaver timeout to *N* seconds, or turns it **off**.

-x   Use hexadecimal digits for output.

Video Mode Support

Note that not all modes listed above may be supported by the video hardware. You can verify which mode is supported by the video hardware, using the **-i mode** option.

The VESA BIOS support must be linked to the kernel or loaded as a KLD module if you wish to use VESA video modes or 132 column modes (see vga(4)).

You need to compile your kernel with the *VGA_WIDTH90* option if you wish to use VGA 90 column modes (see vga(4)).

Video modes other than 25 and 30 line modes may require specific size of font. Use **-f** option above to load a font file to the kernel. If the required size of font has not been loaded to the kernel, **vidcontrol** will fail if the user attempts to set a new video mode.

Modes	Font size
25 line modes	8x16 (VGA), 8x14 (EGA)
30 line modes	8x16
43 line modes	8x8
50 line modes	8x8
60 line modes	8x8

It is better to always load all three sizes (8x8, 8x14 and 8x16) of the same font.

You may set variables in *etc/rc.conf* or *etc/rc.conf.local* so that desired font files will be automatically loaded when the system starts up. See below.

If you want to use any of the raster text modes you need to recompile your kernel with the SC_PIXEL_MODE option. See syscons(4) or vt(4) (depending on which driver you use) for more details on this kernel option.

Format of Video Buffer Dump

The **vidcontrol** utility uses the syscons(4) or vt(4) CONS_SCRSHOT ioctl(2) to capture the current contents of the video buffer. The **vidcontrol** utility writes version and additional information to the standard output, followed by the contents of the video buffer.

VGA video memory is typically arranged in two byte tuples, one per character position. In each tuple, the first byte will be the character code, and the second byte is the character's color attribute.

The VGA color attribute byte looks like this:

bits#		width	meaning
7	<X0000000>	1	character blinking
6:4	<0XXX0000>	3	background color
3	<0000X000>	1	bright foreground color
2:0	<00000XXX>	3	foreground color

Here is a list of the three bit wide base colors:

0	Black
1	Blue

    2   Green

    3   Cyan

    4   Red

    5   Magenta

    6   Brown

    7   Light Grey

Base colors with bit 3 (the bright foreground flag) set:

    0   Dark Grey

    1   Light Blue

    2   Light Green

    3   Light Cyan

    4   Light Red

    5   Light Magenta

    6   Yellow

    7   White

For example, the two bytes

    65 158

specify an uppercase A (character code 65), blinking (bit 7 set) in yellow (bits 3:0) on a blue background (bits 6:4).

The **vidcontrol** output contains a small header which includes additional information which may be useful to utilities processing the output.

The first 10 bytes are always arranged as follows:

Byte Range	Contents
1 - 8	Literal text "SCRSHOT_"
9	File format version number
10	Remaining number of bytes in the header

Subsequent bytes depend on the version number.

Version	Byte	Meaning
1	11	Terminal width, in characters
	12	Terminal depth, in characters

　　　　　　13 and up　　The snapshot data

So a dump of an 80x25 screen would start (in hex)

```
53 43 52 53 48 4f 54 5f 01 02 50 19
---------------------- -- -- -- --
          |          | | | ' 25 decimal
          |          | | '--- 80 decimal
          |          | '------ 2 remaining bytes of header data
          |          '--------- File format version 1
          '----------------------- Literal "SCRSHOT_"
```

VIDEO OUTPUT CONFIGURATION

Boot Time Configuration

You may set the following variables in */etc/rc.conf* or */etc/rc.conf.local* in order to configure the video output at boot time.

blanktime　　　Sets the timeout value for the **-t** option.
font8x16, *font8x14*, *font8x8*
　　　　　　　　Specifies font files for the **-f** option.
scrnmap　　　Specifies a screen output map file for the **-l** option.

See rc.conf(5) for more details.

Driver Configuration

The video card driver may let you change default configuration options, such as the default font, so that you do not need to set up the options at boot time. See video card driver manuals, (e.g., vga(4)) for details.

FILES

/usr/share/syscons/fonts/*
/usr/share/vt/fonts/*　　　　　　　　font files.
/usr/share/syscons/scrnmaps/*　　　screen output map files (relevant for syscons(4) only).

EXAMPLES

If you want to load */usr/share/syscons/fonts/iso-8x16.fnt* to the kernel, run **vidcontrol** as:

```
vidcontrol -f 8x16 /usr/share/syscons/fonts/iso-8x16.fnt
```

So long as the font file is in */usr/share/syscons/fonts* (if using syscons) or */usr/share/vt/fonts* (if using vt),

you may abbreviate the file name as *iso-8x16*:

    vidcontrol -f 8x16 iso-8x16

Furthermore, you can also omit font size "8x16":

    vidcontrol -f iso-8x16

Moreover, the suffix specifying the font size can also be omitted; in this case, **vidcontrol** will use the size of the currently displayed font to construct the suffix:

    vidcontrol -f iso

Likewise, you can also abbreviate the screen output map file name for the **-l** option if the file is found in */usr/share/syscons/scrnmaps*.

    vidcontrol -l iso-8859-1_to_cp437

The above command will load */usr/share/syscons/scrnmaps/iso-8859-1_to_cp437.scm*.

The following command will set-up a 100x37 raster text mode (useful for some LCD models):

    vidcontrol -g 100x37 VESA_800x600

The following command will capture the contents of the first virtual terminal video buffer, and redirect the output to the *shot.scr* file:

    vidcontrol -p < /dev/ttyv0 > shot.scr

The following command will dump contents of the fourth virtual terminal video buffer to the standard output in the human readable format:

    vidcontrol -P < /dev/ttyv3

SEE ALSO

    kbdcontrol(1), vidfont(1), keyboard(4), screen(4), syscons(4), vga(4), vt(4), rc.conf(5), kldload(8), moused(8), watch(8)

    The various *scr2** utilities in the *graphics* and *textproc* categories of the *Ports Collection*.

AUTHORS

Sóren Schmidt *<sos@FreeBSD.org>*
Sascha Wildner *<saw@online.de>*

CONTRIBUTORS

Maxim Sobolev *<sobomax@FreeBSD.org>*
Nik Clayton *<nik@FreeBSD.org>*