VioGetState

Bindings: C, MASM

This call returns the current settings of the palette registers, overscan (border) color, blink/background intensity switch, color registers, underline location, or target VioSetMode display configuration.

VioGetState (RequestBlock, VioHandle)

RequestBlock (**PVOID**) - input/output Address of the video state structures consisting of six different structures depending on the request type:

Type	Definition
0	Get palette registers
1	Get overscan (border) color
2	Get blink/background intensity switch
3	Get color registers
4	Reserved
5	Get the scan line for underlining
6	Get target VioSetMode display configuration.
7	Reserved The six structures, depending on request type, are:

VIOPALSTATE Applies to EGA, VGA, or IBM Personal System/2 Display Adapter.

length (**USHORT**) - input Length of structure, including length.

38	Maximum valid value.

type (**USHORT**) - input Request type 0 for palette registers.

palette (**USHORT**) - input First palette register in the palette register sequence; must be specified in the range 0 through 15. The palette registers are returned in sequential order. The number returned is based upon length.

color (**USHORT** *(length-6)/2) - output Color value for each palette register. The maximum number of entries in the color value array is 16.

VIOOVERSCAN Applies to CGA, VGA, or IBM Personal System/2 Display Adapter.

length (**USHORT**) - input Length of structure, including length.

6	Only valid value.

type (**USHORT**) - input Request type 1 for overscan (border) color.

color (**USHORT**) - input Color value.

VIOINTENSITY Applies to CGA, EGA, MCGA, VGA, or IBM Personal System/2 Display Adapter.

length (**USHORT**) - input Length of structure, including length.

6	Only valid value.

type (**USHORT**) - input Request type 2 for blink/background intensity switch.

switch (**USHORT**) - output **Switch set as**:

Value	**Definition**
0	Blinking foreground colors enabled.
1	High intensity background colors enabled.

VIOCOLORREG Applies to VGA, or IBM Personal System/2 Display Adapter.

length (**USHORT**) - input Length of structure, including length.

12	Length in bytes.

type (**USHORT**) - input Request type 3 for color registers.

first color (**USHORT**) - input First color register to get in the color register sequence; must be specified in the range 0 through 255. The color registers are returned in sequential order.

number color (**USHORT**) - input Number of color registers to get; must be specified in the range 1 through 256.

datarea (**PCH**) - input Far address of a data area where the color registers are returned. The size of the data area must be three bytes times the number of color registers to get. The format of each entry returned is as follows:

Byte 1	Red value
Byte 2	Green value
Byte 3	Blue value

VIOSETULINELOC Applies to EGA, VGA, or IBM Personal System/2 Display Adapter.

length (**USHORT**) - input Length of structure, including length.

6	Length in bytes.

type (**USHORT**) - input Request type 5 to get the scan line for underlining. Underlining is enabled only when the foreground color is 1 or 9.

scanline (**USHORT**) - output The value returned is in the range 0 through 31 and is the scan line minus 1. A value of 32 means underlining is disabled.

VIOSETTARGET

length (**USHORT**) - input Length of structure, including length.

6	Length in bytes.

type (**USHORT**) - input Request type 6 to get display configuration selected to be the target of the next VioSetMode.

select (**USHORT**) - output **Configuration**:

Value	Definition
0	Default selection algorithm. See VioSetMode.
1	Primary
2	Secondary.

VioHandle (**HVIO**) - input Reserved word of 0s.

rc (**USHORT**) - return

Return code descriptions are:

0	NO_ERROR
355	ERROR_VIO_MODE
421	ERROR_VIO_INVALID_PARMS
436	ERROR_VIO_INVALID_HANDLE
438	ERROR_VIO_INVALID_LENGTH
465	ERROR_VIO_DETACHED
494	ERROR_VIO_EXTENDED_SG

Family API Considerations

Request type = 6, Get Target VioSetMode Display Configuration, and request type = 5, Get Underline Location, are not supported in the family API.