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

Syntax

VioGetState (RequestBlock, VioHandle)

Parameters

;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. 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. Only valid value. :type (USHORT) - input: Request type 2 for blink/background intensity switch. :switch (USHORT) - output: Switch set as: ::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. 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. 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: ::0 Default selection algorithm. See VioSetMode. ::1 Primary ::2 Secondary ;VioHandle (HVIO) - input : Reserved word of 0s.

Return Code

;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

Remarks

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.

Bindings

C

<PRE> typedef struct _VIOPALSTATE {

```
USHORT  cb;                     /* Length of this structure in bytes */
USHORT  type;                   /* Request type=0 get palette registers */
USHORT  iFirst;                 /* First palette register to return */
USHORT  acolor[1];              /* Color value palette register */
}VIOPALSTATE;
```

typedef VIOPALSTATE far *PVIOPALSTATE;

typedef struct _VIOOVERSCAN {

```
USHORT  cb;                     /* Length of this structure */
USHORT  type;                   /* Request type=1 get overscan
                                    (border) color */
USHORT  color;                  /* Color value */
}VIOOVERSCAN;
```

typedef VIOOVERSCAN far *PVIOOVERSCAN;

typedef struct _VIOINTENSITY {

```
USHORT  cb;                     /* Length of this structure */
USHORT  type;                   /* Request type=2 get blink/background
```

```
                                    intensity switch */
USHORT  fs;                    /* Value of blink/background switch */
}VIOINTENSITY;
```

typedef VIOINTENSITY far *PVIOINTENSITY;

typedef struct _VIOCOLORREG { /* viocreg */

```
USHORT  cb;
USHORT  type;
USHORT  firstcolorreg;
USHORT  numcolorregs;
PCH     colorregaddr;
}VIOCOLORREG;
```

typedef VIOCOLORREG far *PVIOCOLORREG;

typedef struct _VIOSETULINELOC { /* viouline */

```
USHORT  cb;
USHORT  type;
USHORT  scanline;
}VIOSETULINELOC;
```

typedef VIOSETULINELOC far *PVIOSETULINELOC;

typedef struct _VIOSETTARGET { /* viosett */

```
USHORT  cb;
USHORT  type;
USHORT  defaultalgorithm;
}VIOSETTARGET;
```

typedef VIOSETTARGET far *PVIOSETTARGET;

#define INCL_VIO

USHORT rc = VioGetState(RequestBlock, VioHandle);

PVOID RequestBlock; /* Request block */ HVIO VioHandle; /* Vio handle */

USHORT rc; /* return code */ </PRE>

MASM

<PRE> VIOPALSTATE struc

```
viopal_cb               dw ? ;Length of this structure in bytes
viopal_type             dw ? ;Request type=0 get palette registers
viopal_iFirst           dw ? ;First palette register to return
```

```
viopal_acolor  dw 1  dup (?) ;Color value palette register
```

VIOPALSTATE ends

VIOOVERSCAN struc

```
vioos_cb                dw ? ;Length of this structure
vioos_type              dw ? ;Request type=1 get overscan (border) color
vioos_color             dw ? ;Color value
```

VIOOVERSCAN ends

VIOINTENSITY struc

```
vioint_cb               dw ? ;Length of this structure
vioint_type             dw ? ;Request type=2 get blink/background
                             ; intensity switch
vioint_fs               dw ? ;Value of blink/background switch
```

VIOINTENSITY ends

VIOCOLORREG struc

```
viocreg_cb              dw ? ;
viocreg_type            dw ? ;
viocreg_firstcolorreg   dw ? ;
viocreg_numcolorregs    dw ? ;
viocreg_colorregaddr    dd ? ;
```

VIOCOLORREG ends

VIOSETULINELOC struc

```
viouline_cb             dw ? ;
viouline_type           dw ? ;
viouline_scanline       dw ? ;
```

VIOSETULINELOC ends

VIOSETTARGET struc

```
viosett_cb               dw ? ;
viosett_type             dw ? ;
viosett_defaultalgorithm dw ? ;
```

VIOSETTARGET ends

EXTRN VioGetState:FAR INCL_VIO EQU 1

PUSH@ OTHER RequestBlock ;Request block PUSH WORD VioHandle ;Vio handle CALL VioGetState

Returns WORD </PRE>

Vio