

This call sets the mode of the display.

## Syntax

VioSetMode (ModeData, VioHandle)

## Parameters

;ModeData ([PVIOMODEINFO](#)) - input : Address of the mode characteristics structure: ::length (USHORT) : Input parameter to VioSetMode. Length specifies the length of the data structure in bytes including Length itself. The minimum structure size required is 3 bytes. OS/2 sets to the first mode (in the list of modes supported by this display configuration) with a data structure matching the mode data specified. ::type (UCHAR): Mode characteristics bit mask: 'Bit Description' 7-4 Reserved, set to zero. 3 0 = VGA-compatible modes 0 thru 13H.

1 = Native mode.

2 0 = Enable color burst

1 = Disable color burst.

1 0 = Text mode.

1 = Graphics mode.

0 0 = Monochrome compatible mode.

1 = Other.

::numcolors (UCHAR): Number of colors defined as a power of 2. This is equivalent to the number of color bits that define the color, for example: 'Value Definition' 0 Monochrome modes 7, 7+, and F. 1 2 colors. 2 4 colors. 4 16 colors. 8 256 colors. ::textcols (USHORT): Number of text columns.

::textrows (USHORT): Number of text rows. ::pelcols (USHORT): Horizontal resolution, number of pel columns. ::pelrows (USHORT): Vertical resolution, number of pel rows. ::Attribute Format (UCHAR): Identifies the format of the attributes. ::Number of Attributes (UCHAR): Identifies the number of attributes in a character cell. ::Buffer Address (ULONG): 32-bit physical address of the physical display buffer for this mode. ::Buffer Length (ULONG): Length of the physical display buffer for this mode. ::Full Buffer Size (ULONG): Size of the buffer required for a full save of the physical display buffer for this mode. ::Partial Buffer Size (ULONG): Size of the buffer required for a partial (pop-up) save of the physical display buffer for this mode. ::Extended Data Area Address (PCH): Far address to an extended mode data structure or zero if none. The format of the extended mode data structure is determined by the device driver and is unknown to OS/2. ;VioHandle (HVIO) - input : Reserved word of 0s.

## Return Code

rc (USHORT) - return Return code descriptions are: \*0 NO\_ERROR \*355 ERROR\_VIO\_MODE \*430

EMULATED							
ERROR_VIO_ILLEGAL_DURING_POPUP *436 ERROR_VIO_INVALID_HANDLE *438							
ERROR_VIO_INVALID_TYPE *460 ERROR_VIO_DETACHED *467 ERROR_VIO_FONT *468							
ERROR_VIO_WRONG_TYPE *469 ERROR_VIO_EXTENDED_SG							
0+	5	4	40	25	360	400	
Remarks	5	4	40	25	320	400	
1	1	4	40	25	320	200	
VioSetMode	1	4	40	25	320	350	position and type.
1+	1	4	40	25	360	400	VioSetMode does not clear the screen. To clear the screen, use one of the VioScrollxx calls.
1#	1	4	40	25	320	400	The disable burst bit in the Type field in the VioSetMode data structure is functional only for the CGA and VGA. This bit causes the color portion of the video signal to be suppressed, producing a black and white mode on composite monitors attached to the CGA. On VGA, the bit causes the color lookup table to be loaded with values that produce shades of gray instead of colors, again producing a black and white mode. For all other combinations of adapters and displays, the setting of this bit is recorded and returned on any subsequent VioGetMode call, but otherwise is ignored.
2	50	4	80	25	640	200	
2*	5	4	80	25	640	350	
3	5	4	80	25	720	400	
3*	5	4	80	25	640	400	
4	5	4	80	25	640	350	
4*	5	4	80	25	720	350	
5	5	4	80	25	640	200	
5*	5	4	80	25	640	200	
6	5	4	80	25	640	350	
6*	5	4	80	25	720	350	
7	0	0	80	25	720	400	
7#	0	0	80	25	640	400	If VioSetState request type 6 has been issued previously to select the target display configuration for VioSetMode, the mode is set on the display configuration selected. If that display configuration does not support the mode specified, an error is returned.
n/a	1	4	80	30	720	480	
n/a	1	4	80	30	640	480	Assuming no target display configuration for VioSetMode is selected, the mode is set on the primary configuration. If the primary configuration does not support the mode specified, the mode is set on the secondary.
8	1	4	80	25	640	200	
9	1	4	80	25	640	200	
10	3	4	[80]	[25]	640	350	
11	3	1	[80]	[30]	640	480	
12	3	4	[80]	[30]	640	480	
13	3	8	[40]	[25]	320	200	
n/a	11	8	[80]	[30]	640	480	
n/a	11	4	[80]	[30]	640	480	
n/a	11	8	[85]	[38]	1024	768	
n/a	11	4	[85]	[38]	1024	768	
colspan=8							

MPA Monochrome/Printer Adapter CGA Color Graphics Adapter EGA Enhanced Graphics Adapter VGA

## Video Graphics Array, PS/2 Display Adapter 8514A 8514/A Display Adapter

### PM Considerations

colspan=8

Windows Monochrome Display, 8515 Color Display, EGA 5154 Enhanced Color Display, MONO 8503 B&W Monochrome Display, 8507/8604 Display, HMONO 8507/8604 Display, CGA 8512/13 PS/2 Color Display, 8514 Display, 256 COLOR 8514 Display, PLASMA Plasma Display Using COMPOSITE Video Monitor 1, Color = 4, Text Columns = 80, Text Rows = requested Text Rows, Horizontal Resolution = 640, and Vertical Resolution = 16 \* (Text Rows).

Notes: #Types 0, 1, and 5 are text modes; types 2, 3, 7, and 11 are graphics modes. #For BIOS modes 0, 2, 5, the color burst is disabled on the CGA and VGA. #The Personal System/2 Display Adapter API Considerations

function modes, which are supported through the 8514/A display adapter interface, not the VIO Subsystem. Refer to the Personal System/2 Display Adapter 8514/A Technical Reference for details of this support.

Some options operate differently in the DOS mode than in the OS/2 mode. Therefore, the following consideration applies to VioSetMode when coding for the DOS mode:

VioSetMode clears the screen.

### Bindings

## C Binding

```
<PRE> typedef struct _VIOMODEINFO {
```

```
USHORT cb; /* Length of the entire data structure */
UCHAR fbType; /* Bit mask of mode being set */
UCHAR color; /* Number of colors (power of 2) */
USHORT col; /* Number of text columns */
USHORT row; /* Number of text rows */
USHORT hres; /* Horizontal resolution */
USHORT vres; /* Vertical resolution */
UCHAR fmt_ID; /* Attribute format */
UCHAR attrib; /* Number of attributes */
ULONG buf_addr;
ULONG buf_length;
ULONG full_length;
ULONG partial_length;
PCH ext_data_addr;
} VIOMODEINFO;
```

```
typedef VIOMODEINFO far *PVIOMODEINFO;
```

```
#define INCL_VIO
```

```
USHORT rc = VioSetMode(ModeData, VioHandle);
```

```
PVIOMODEINFO ModeData; /* Mode characteristics */ HVIO VioHandle; /* Video handle */
```

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

## MASM Binding

<PRE> VIOMODEINFO struc

```
viomi_cb          dw ? ;Length of the entire data structure
viomi_fbType      db ? ;Bit mask of mode being set
viomi_color       db ? ;Number of colors (power of 2)
viomi_col         dw ? ;Number of text columns
viomi_row          dw ? ;Number of text rows
viomi_hres         dw ? ;Horizontal resolution
viomi_vres         dw ? ;Vertical resolution
viomi_fmt_ID       db ? ;Attribute format
viomi_attrib       db ? ;Number of attributes
viomi_buf_addr     dd ? ;
viomi_buf_length   dd ? ;
viomi_full_length dd ? ;
viomi_partial_length dd ? ;
viomi_ext_data_addr dd ? ;
```

VIOMODEINFO ends

EXTRN VioSetMode:FAR INCL\_VIO EQU 1

PUSH@ OTHER ModeData ;Mode characteristics PUSH WORD VioHandle ;Video handle CALL  
VioSetMode

Returns WORD </PRE>

## Note

Text based on [http://www.edm2.com/index.php/VioSetMode\\_\(FAPI\)](http://www.edm2.com/index.php/VioSetMode_(FAPI))

Family API		
DOS	Process Manager	DosBeep DosExit DosSleep DosExecPgm
	File Manager	DosChDir DosChgFilePtr DosClose DosDelete DosDupHandle DosMkDir DosMove DosQCurDir DosQCurDisk DosSet FileMode DosOpen DosQFileInfo DosRead DosQFileInfo DosQFSInfo DosQVerify DosRmDir DosSelectDisk DosFindClose DosFindFirst DosFindNext DosSet FileInfo DosSet Verify DosWrite DosFileLocks DosSet FHandState DosNewSize DosBufReset DosQFHandState DosSet FInfo
	Memory Manager	DosFreeSeg DosSubAlloc DosSubFree DosSubSet DosAlloc Huge DosAlloc Seg DosRealloc Huge DosRealloc Seg DosGet Huge Shift DosCreateCS Alias
	NLS	DosCaseMap DosGetCtryInfo DosGetDBCSEv DosSetCtryCode DosGetCollate DosGetMessage DosInsMessage DosPutMessage
	Date and Time	DosSetDateTime DosGetDateTime
	Devices	DosDevConfig DosDevIOCtl DosDevIOCtl2
	Signals	DosHoldSignal DosSetSigHandler
	Misc	BadDynLink DosGetEnv DosGetMachineMode DosGetVersion DosError DosErrClass DosSetVec
KBD		KbdCharIn KbdFlushBuffer KbdGetStatus KbdSetStatus KbdStringIn KbdPeek
VIO		VioGetBuf VioGetConfig VioGetCurPos VioGetCurType VioGetPhysBuf VioReadCellStr VioReadCharStr VioScrollUp VioScrollDn VioScrollLf VioScrollRt VioScrUnLock VioSetCurPos VioSetCurType VioSetMode VioGetMode VioShowBuf VioWrtCellStr VioWrtCharStr VioWrtCharStrAtt VioWrtNAttr VioWrtNCell VioWrtNChar VioWrtTTY VioScrLock VioPopUp
Tools		BIND
Modules		DOSCALLS.DLL VIOCALS.DLL KBDCALLS.DLL MSG.DLL
Libraries		API.LIB OS2386.LIB FAPI.LIB DOSCALLS.LIB SUBCALLS.LIB

2018/08/25 15:05 · prokushev · 0 Comments

From:  
<https://ftp.osfree.org/doku/> - osFree wiki



Permanent link:  
<https://ftp.osfree.org/doku/doku.php?id=en:docs:fapi:viosetmode&rev=1535796301>

Last update: 2018/09/01 10:05