

## VioGetMode

**Bindings:** C, MASM

This call returns the mode of the display.

*VioGetMode* (ModeData, VioHandle)

*ModeData* (**PVIOMODEINFO**) - input/output Far address of a structure where mode characteristics are returned.

*length* (**USHORT**) Input parameter to *VioGetMode*. *Length* specifies the length of the data structure in bytes including *length* itself. The value specified on input controls the amount of mode data returned. The minimum structure size required is 2 bytes, and the maximum structure size required is 34 bytes. For OS/2 1.2, a *length* of 2 returns the size of the maximum structure required for all the mode data. When *length* is not equal to 2, the *length* field is modified on output to reflect the actual number of bytes returned.

*type* (**UCHAR**) Mode characteristics bit mask:

Bit	Description
7-4	Reserved
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)** Format of the attributes.

**Number of Attributes (UCHAR)** 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.

**rc (USHORT)** - return Return code descriptions are:

0	NO_ERROR
436	ERROR_VIO_INVALID_HANDLE
438	ERROR_VIO_INVALID_LENGTH
465	ERROR_VIO_DETACHED
494	ERROR_VIO_EXTENDED_SG

## Remarks

Refer to [VioSetMode](#) for examples.

## C bindings

```
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 = VioGetMode(ModeData, VioHandle);

PVIOMODEINFO ModeData; /* Mode characteristics */
HVIO VioHandle; /* Vio handle */

USHORT rc; /* return code */

```

## MASM bindings

```

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 VioGetMode:FAR
INCL_VIO EQU 1

PUSH@ OTHER ModeData ;Mode characteristics
PUSH WORD VioHandle ;Vio handle
CALL VioGetMode

Return WORD

```

From:

<https://osfree.ru/doku/> - **osFree wiki**

Permanent link:

<https://osfree.ru/doku/doku.php?id=en:ibm:prcp:vio:getmode>

Last update: **2016/09/15 05:11**

