



This is part of **Family API** which allow to create dual-os version of program runs under OS/2 and DOS

**Note:** This is legacy API call. It is recommended to use 32-bit equivalent

2021/09/17 04:47 · prokushev · [0 Comments](#)

2021/08/20 03:18 · prokushev · [0 Comments](#)

## VioModeWait

This call allows a graphics mode application to be notified when it must restore its video mode, state, and modified display adapter registers. The return from this function call provides the notification.

### Syntax

```
VioModeWait (RequestType, NotifyType, Reserved)
```

### Parameters

- RequestType (USHORT) - input : Application request event. RequestType = 0 indicates the application wants to be notified at the end of a pop-up to restore its mode. RequestType = 0 is the only event supported by VioModeWait.
- NotifyType (PUSHORT) - output : Address of the operation to be performed by the application returning from VioModeWait. NotifyType = 0, indicating restore mode, is the only type of notification returned.
- Reserved (USHORT) - input : Reserved word of 0s.

### Return Code

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

- 0 NO\_ERROR
- 421 ERROR\_VIO\_INVALID\_PARMS
- 422 ERROR\_VIO\_FUNCTION\_OWNED
- 423 ERROR\_VIO\_RETURN
- 424 ERROR\_SCS\_INVALID\_FUNCTION
- 428 ERROR\_VIO\_NO\_SAVE\_RESTORE\_THD
- 430 ERROR\_VIO\_ILLEGAL\_DURING\_POPUP
- 465 ERROR\_VIO\_DETACHED
- 494 ERROR\_VIO\_EXTENDED\_SG

## Remarks

At the completion of an application or hard error pop-up (reference `VioPopUp`), OS/2 notifies the session that was originally interrupted for the pop-up to restore its mode. The return from this function call provides that notification. The thread that issued the call must perform the restore and then immediately re-issue `VioModeWait`.

When an application's `VioModeWait` thread is notified, the thread must restore its video mode, state, and modified display adapter registers. An application's `VioModeWait` thread does not restore the physical display buffer. OS/2 saves/restores the physical display buffer over a pop-up.

Only one process for a session can issue `VioModeWait`. The first process that issues `VioModeWait` becomes the owner of this function. (Refer to `VioModeUndo`.)

An application must issue `VioModeWait` only if it writes directly to the registers on the display adapter. Otherwise, the application can allow OS/2 to perform the required restore by not issuing `VioModeWait`.

When an application issues `VioModeWait`, it is also required to issue `VioSavRedrawWait` to be notified at screen switch time to perform a full save or restore (reference `VioSavRedrawWait`). Two application threads must be dedicated to performing these operations.

## Bindings

### C

```
#define INCL_VIO

USHORT rc = VioModeWait(RequestType, NotifyType, Reserved);

USHORT RequestType; /* Request type */
USHORT NotifyType; /* Notify type (returned) */
USHORT Reserved; /* Reserved (must be zero) */

USHORT rc; /* return code */
```

### MASM

```
EXTRN VioModeWait:FAR
INCL_VIO EQU 1

PUSH WORD RequestType ;Request type
PUSH@ WORD NotifyType ;Notify type (returned)
PUSH WORD Reserved ;Reserved (must be zero)
CALL VioModeWait
```

Returns **WORD**

[http://www.edm2.com/index.php/VioModeWait\\_\(OS/2\\_1.x\)](http://www.edm2.com/index.php/VioModeWait_(OS/2_1.x))

From:

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

Permanent link:

<https://www.osfree.org/doku/doku.php?id=en:docs:fapi:viomodewait&rev=1634194533>

Last update: **2021/10/14 06:55**

