 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

# VioSavRedrawWait

This call notifies a graphics mode application when it must save or redraw its screen image.

## Syntax

```
VioSavRedrawWait (SavRedrawIndic, NotifyType, VioHandle)
```

## Parameters

- SavRedrawIndic (USHORT) - input : Indicates which events the application is waiting for:
  - 0 - The session manager notifies the application for both save and redraw operations.
  - 1 - The session manager notifies the application for redraw operations only.
- NotifyType (PUSHORT) - output : Address that specifies the operation to be performed by the application upon return from VioSavRedrawWait:
  - 0 - Save screen image
  - 1 - Restore screen image.
- VioHandle (HVIO) - 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
- 430 ERROR_VIO_ILLEGAL_DURING_POPUP
- 436 ERROR_VIO_INVALID_HANDLE
- 465 ERROR_VIO_DETACHED
- 494 ERROR_VIO_EXTENDED_SG

# Remarks

OS/2 uses VioSavRedrawWait to notify a graphics mode application to save or restore its screen image at screen switch time. The application in the outgoing foreground session is notified to perform a save. The application in the incoming foreground session is notified to perform a restore. The application must perform the action requested and immediately re-issue VioSavRedrawWait. When an application performs a save, it saves its physical display buffer, video mode, and any other information the application needs to completely redraw its screen at restore time.

Only one process per session can issue VioSavRedrawWait. The process that first issues VioSavRedrawWait becomes the owner of the function.

A text mode application must issue VioSavRedrawWait only if the application writes directly to the registers on the display adapter. Assuming VioSavRedrawWait is not issued by a text mode application, OS/2 performs the required saves and restores.

An application that issues VioSavRedrawWait may also need to issue VioModeWait. This would allow the application to be notified when it must restore its mode at the completion of an application or hard error pop-up. Refer to VioModeWait for more information. Two application threads would be required to perform these operations in this case.

At the time a VioSavRedrawWait thread is notified, the session is in transition to/from the background. Although the session's official status is background, any selector to the physical display buffer previously obtained by the VioSavRedrawWait process (through VioGetPhysBuf) is valid at this time. The physical display buffer must be accessed without issuing VioScrLock. Since the session's official status is background, any thread waits if it issues VioScrLock with the “wait if unsuccessful” option.

An application containing a VioSavRedrawWait thread should be designed so that the process does not cause any hard errors while the VioSavRedrawWait thread is running, otherwise a system lockout may occur.

An application's VioSavRedrawWait thread may be notified to perform a restore before it is notified to perform a save. This happens if the application was running in the background the first time it issued VioSavRedrawWait. The return from this function call provides the notification. The thread that issues the call performs the save or redraw and then reissues VioSavRedrawWait to wait until its screen image must be saved or redrawn again.

# Bindings

## C

```
#define INCL_VIO

USHORT  rc = VioSavRedrawWait(SavRedrawIndic, NotifyType, VioHandle);

USHORT  SavRedrawIndic; /* Save/redraw indicator */
PUSHORT NotifyType;     /* Notify type (returned) */
HVIO    VioHandle;      /* Video handle */
```

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

## MASM

```
EXTRN  VioSavRedrawWait:FAR
INCL_VIO            EQU 1

PUSH   WORD    SavRedrawIndic ;Save/redraw indicator
PUSH@  WORD    NotifyType     ;Notify type (returned)
PUSH   WORD    VioHandle      ;Video handle
CALL   VioSavRedrawWait

Returns WORD
```

http://www.edm2.com/index.php/VioSavRedrawWait_(OS/2_1.x)

| **Family API** | | |
|---|---|---|
| DOS | Process Manager | DosBeep DosExit DosSleep DosExecPgm |
| | File Manager | DosChDir DosChgFilePtr DosClose DosDelete DosDupHandle DosMkDir DosMove DosQCurDir DosQCurDisk DosSetFileMode DosOpen DosQFileInfo DosRead DosQFileMode DosQFSInfo DosQVerify DosRmDir DosSelectDisk DosFindClose DosFindFirst DosFindNext DosSetFileInfo DosSetVerify DosWrite DosFileLocks DosSetFHandState DosNewSize DosBufReset DosQFHandState DosSetFSinfo DosShutdown |
| | Memory Manager | DosFreeSeg DosSubAlloc DosSubFree DosSubSet DosAllocHuge DosAllocSeg DosReallocHuge DosReallocSeg DosGetHugeShift DosCreateCSAlias |
| | 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 VIOCALLS.DLL KBDCALLS.DLL MSG.DLL |
| Libraries | | API.LIB OS2386.LIB FAPI.LIB DOSCALLS.LIB SUBCALLS.LIB |

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