This call allows one thread within a process to cancel a VioModeWait issued by another thread within the same process.

Syntax

VioModeUndo (OwnerIndic, KillIndic, Reserved)

Parameters

;OwnerIndic (USHORT) - input : Indicates whether the thread issuing VioModeUndo wants ownership of VioModeWait to be reserved for its process. ::0 - Reserve ownership ::1 - Give up ownership ;KillIndic (USHORT) - input : Indicates whether the thread (with the outstanding VioModeWait) should be returned an error code or be terminated. ::0 - Return error code ::1 - Terminate thread ;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 *427 ERROR_VIO_NO_MODE_THREAD *430 ERROR_VIO_ILLEGAL_DURING_POPUP *465 ERROR_VIO_DETACHED *486 ERROR_VIO_BAD_RESERVE *494 ERROR_VIO_EXTENDED_SG

Remarks

VioModeUndo may be issued only by a thread within the process that owns VioModeWait. The thread issuing VioModeUndo can either reserve ownership of the VioModeWait function for its process or give up ownership. The thread whose VioModeWait is cancelled is optionally terminated.

Bindings

C

<PRE> #define INCL_VIO

USHORT rc = VioModeUndo(OwnerIndic, KillIndic, Reserved);

USHORT OwnerIndic; /* Ownership indicator */ USHORT KillIndic; /* Terminate indicator */ USHORT Reserved; /* Reserved (must be zero) */

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

MASM

<PRE> EXTRN VioModeUndo:FAR INCL_VIO EQU 1

PUSH WORD OwnerIndic ;Ownership indicator PUSH WORD KillIndic ;Terminate indicator PUSH WORD Reserved ;Reserved (must be zero) CALL VioModeUndo

Returns WORD </PRE>

http://www.edm2.com/index.php/VioModeUndo_(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:viomodeundo&rev=1630817906

Last update: **2021/09/05 04:58**