

VioModeUndo

Bindings: C, MASM

This call allows one thread within a process to cancel a [VioModeWait](#) issued by another thread within the same process.

VioModeUndo (OwnerIndic, KillIndic, Reserved)

OwnerIndic (**USHORT**) - input Indicates whether the thread issuing [VioModeUndo](#) wants ownership of [VioModeWait](#) to be reserved for its process.

Value	Definition
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.

Value	Definition
0	Return error code
1	Terminate thread.

Reserved (**USHORT**) - input Reserved word of 0s.

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.

C bindings

```
#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 */
```

MASM bindings

```
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
```

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

Permanent link:
<https://ftp.osfree.org/doku/doku.php?id=en:ibm:prcp:modeundo>

Last update: **2016/09/15 05:41**

