 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

DosDevIOCtl

This call performs control functions on a device specified by an opened device handle.

Syntax

```
DosDevIOCtl (Data, ParmList, Function, Category, DevHandle)
```

Parameters

- Data (PVOID) - input : Address of the data area.
- ParmList (PVOID) - input : Address of the command-specific argument list.
- Function (USHORT) - input : Device-specific function code.
- Category (USHORT) - input : Device category.
- DevHandle (HFILE) - input : Device handle returned by DosOpen or a standard (open) device handle.

Return Code

rc (USHORT) - return

Return code descriptions are:

- 0 NO_ERROR
- 1 ERROR_INVALID_FUNCTION
- 6 ERROR_INVALID_HANDLE
- 15 ERROR_INVALID_DRIVE
- 31 ERROR_GEN_FAILURE
- 87 ERROR_INVALID_PARAMETER
- 115 ERROR_PROTECTION_VIOLATION
- 117 ERROR_INVALID_CATEGORY
- 119 ERROR_BAD_DRIVER_LEVEL
- 163 ERROR_UNCERTAIN_MEDIA
- 165 ERROR_MONITORS_NOT_SUPPORTED

Remarks

Values returned in the range hex FF00 through FFFF are user dependent error codes. Values returned in the range hex FE00 through FEFF are device driver dependent error codes.

Refer to the `IBM Operating System/2 Version 1.2 I/O Subsystems And Device Support Volume 1` for a complete listing of control functions (DevHlp calls).

Family API Considerations

Some options operate differently in the DOS mode than in the OS/2 mode. Therefore, the following considerations apply to DosDevIOCtl when coding for the DOS mode.

The level of support for DosDevIOCtl is identified by category and function code with a noted restriction if it is not supported by DOS 2.X or DOS 3.X Functions tend to be more restrictive in lower version numbers of DOS.

- Category 1 supported as follows:
  - 41H Set Baud Rate
  - 42H Set Line Control
  - All other category 1 functions are not supported for DOS 2.X and DOS 3.X.
- Category 2 not supported in FAPI
- Category 3 not supported in FAPI
- Category 4 not supported in FAPI
- Category 5 supported in FAPI as follows:
  - 42H Set Frame control - supports IBM Graphics Printers only
  - 44H Set Infinite Retry - for DOS 2.X and DOS 3.X, the function is in effect only for the duration of the calling program
  - 46H Initialize printer
  - 62H Get Frame Control - not supported for DOS 2.X and DOS 3.X
  - 64H Get Infinite Retry
  - 66H Get Printer Status.
- Category 6 not supported in FAPI
- Category 7 not supported in FAPI
- Category 8 supported in FAPI as follows:
  - 00H Lock Drive - not supported for versions below DOS 3.2
  - 01H Unlock Drive - not supported for versions below DOS 3.2
  - 02H Redetermine Media - not supported for versions below DOS 3.2
  - 03H Set Logical Map - not supported for versions below DOS 3.2
  - 20H Block Removable - not supported for versions below DOS 3.2
  - 21H Get Logical Map - not supported for versions below DOS 3.2
  - 43H Set Device Parameters - not supported for DOS 2.X and DOS 3.X
  - 44H Write Track - not supported for DOS 2.X and DOS 3.X
  - 45H Format Track - not supported for DOS 2.X and DOS 3.X
  - 63H Get Device Parameters - not supported for DOS 2.X and DOS 3.X
  - 64H Read Track - not supported for DOS 2.X and DOS 3.X
  - 65H Verify Track - not supported for DOS 2.X and DOS 3.X.
- Category 9 is reserved
- Category 10 (0AH) not supported in FAPI

- Category 11 (0BH) not supported in FAPI.

Bindings

C Binding

```
#define INCL_DOSDEVICES

USHORT  rc = DosDevIOCtl(Data, ParmList, Function, Category, DevHandle);

PVOID            Data;          /* Data area */
PVOID            ParmList;      /* Command arguments */
USHORT           Function;      /* Device function */
USHORT           Category;      /* Device category */
HFILE            DevHandle;     /* Specifies the device */

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

MASM Binding

```
EXTRN  DosDevIOCtl:FAR
INCL_DOSDEVICES     EQU 1

PUSH@  OTHER   Data          ;Data area
PUSH@  OTHER   ParmList      ;Command arguments
PUSH   WORD    Function      ;Device function
PUSH   WORD    Category      ;Device category
PUSH   WORD    DevHandle     ;Device handle
CALL   DosDevIOCtl
```

Returns WORD

Note

Text based on http://www.edm2.com/index.php/DosDevIOCtl_(FAPI)

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