 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

KbdSetStatus

This call sets the characteristics of the keyboard.

Syntax

KbdSetStatus (StatData, KbdHandle)

Parameters

;StatData (PKBDINFO) - input : Address of the keyboard status structure: :length (USHORT) : Length, in bytes, of this data structure, including length. ::10 - Only valid value. :sysstate (USHORT) : The system state altered by this call. If bits 0 and 1 are off, the echo state of the system is not altered. If bits 2 and 3 are off, the binary and ASCII state of the system is not altered. If bits 0 and 1 are on, or if bits 2 and 3 are on, the function returns an error. If binary mode is set, echo is ignored. 'Bit Description' 15-9 Reserved, set to zero 8 Shift return is on 7 Length of the turn-around character (meaningful only if bit 6 is on). 6 Turn-around character is modified 5 Interim character flags are modified 4 Shift state is modified 3 ASCII mode is on 2 Binary mode is on 1 Echo off 0 Echo on :turnchardef (USHORT) : Definition of the turn-around character. In ASCII and extended-ASCII format, the turn-around character is defined as the carriage return. In ASCII format only, the turn-around character is defined in the low-order byte. :intcharflag (USHORT) : Interim character flags: 'Bit Description' 15-8 NLS shift state. 7 Interim character flag is on 6 Reserved, set to zero 5 Application requested immediate conversion 4-0 Reserved, set to zero :shiftstate (USHORT) : Shift state. 'Bit Description' 15 SysReq key down 14 CapsLock key down 13 NumLock key down 12 ScrollLock key down 11 Right Alt key down 10 Right Ctrl key down 9 Left Alt key down 8 Left Ctrl key down 7 Insert on 6 CapsLock on 5 NumLock on 4 ScrollLock on 3 Either Alt key down 2 Either Ctrl key down 1 Left Shift key down 0 Right Shift key down ;KbdHandle (HKBD) - input : Default keyboard or the logical keyboard.

Return Code

rc (USHORT) - return Return code descriptions are: * 0 NO_ERROR * 376 ERROR_KBD_INVALID_LENGTH * 377 ERROR_KBD_INVALID_ECHO_MASK * 378 ERROR_KBD_INVALID_INPUT_MASK * 439 ERROR_KBD_INVALID_HANDLE * 445 ERROR_KBD_FOCUS_REQUIRED * 447 ERROR_KBD_KEYBOARD_BUSY * 464 ERROR_KBD_DETACHED *

504 ERROR_KBD_EXTENDED_SG

Remarks

Shift return (bit 8 in sysstate) must be disabled in ASCII mode.

KbdSetStatus is ignored for a Vio-windowed application.

Family API Considerations

Some options operate differently in the DOS mode than in the OS/2 mode. Therefore, the following restrictions apply to KbdSetStatus when coding in the DOS mode: * KbdSetStatus does not accept a bit mask of 10 (ASCII on, Echo Off). * Raw (binary) Mode and Echo On are not supported and return an error if requested. * KbdHandle is ignored. * Interim character is not supported. * Turnaround character is not supported.

Example Code

C Binding

<PRE> typedef struct _KBDINFO { /* kbst */

```
USHORT cb;                  /* length in bytes of this structure */
USHORT fsMask;              /* bit mask of functions to be altered */
USHORT chTurnAround;        /* define TurnAround character */
USHORT fsInterim;           /* interim character flags */
USHORT fsState;             /* shift states */
```

}KBDINFO;

#define INCL_KBD

USHORT rc = KbdSetStatus(Structure, KbdHandle);

PKBDINFO Structure; /* Data structure */ HKBD KbdHandle; /* Keyboard Handle */

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

MASM Binding

<PRE> KBDINFO struc

```
kbst_cb           dw  ? ;length in bytes of this structure
kbst_fsMask       dw  ? ;bit mask of functions to be altered
kbst_chTurnAround dw  ? ;define TurnAround character
```

```
kbst_fsInterim    dw  ? ;interim character flags
kbst_fsState      dw  ? ;shift states
```

KBDINFO ends

EXTRN KbdSetStatus:FAR INCL_KBD EQU 1

PUSH@ OTHER Structure ;Data structure PUSH WORD KbdHandle ;Keyboard Handle CALL KbdSetStatus

Returns WORD </PRE>

Note

Text based on http://www.edm2.com/index.php/KbdSetStatus_(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