 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

KbdStringIn

This call reads a character string (character codes only) from the keyboard.

Syntax

KbdStringIn (CharBuffer, StringLength, IOWait, KbdHandle)

Parameters

;CharBuffer (PCH) - output : Address of the character string buffer. ;StringLength (PSTRINGINBUF) - input/output : Address of the length of the character string buffer. On entry, buflen is the maximum length, in bytes, of the buffer. The maximum length that can be specified is 255. Template processing has meaning only in the ASCII mode. ::buflen (USHORT) : Length of the input buffer. ::inputlen (USHORT) : Number of bytes read into the buffer. ;IOWait (USHORT) - input : Wait if a character is not available. ::0 - Wait. In Binary input mode, the requestor waits until CharBuffer is full. In ASCII input mode, the requestor waits until a carriage return is pressed. ::1 - No wait. The requestor gets an immediate return if no characters are available. If characters are available, KbdStringIn returns immediately with as many characters as are available (up to the maximum). No wait is not supported in ASCII input mode. ;KbdHandle (HKBD) - input : Default keyboard or the logical keyboard.

Return Code

rc (USHORT) - return Return code descriptions are: * 0 NO_ERROR * 375 ERROR_KBD_INVALID_IOWAIT * 439 ERROR_KBD_INVALID_HANDLE * 445 ERROR_KBD_FOCUS_REQUIRED * 464 ERROR_KBD_DETACHED * 504 ERROR_KBD_EXTENDED_SG

Remarks

The character strings may be optionally echoed on the display if echo mode is set. When echo is on each character is echoed as it is read from the keyboard. Echo mode and BINARY mode are mutually exclusive. Reference KbdSetStatus and KbdGetStatus for more information.

The default input mode is ASCII. In ASCII mode, 2-byte character codes only return in complete form.

An extended ASCII code is returned in a 2-byte string. The first byte is 0DH or E0H and the next byte is an extended code.

In input mode (BINARY, ASCII), The following returns can be set and retrieved with KbdSetStatus and KbdGetStatus: * Turnaround Character * Echo Mode * Interim Character Flag * Shift State

The received input length is also used by the KbdStringIn line edit functions for re-displaying and entering a caller specified string. On the next KbdStringIn call the received input length indicates the length of the input buffer that may be recalled by the user using the line editing keys. A value of 0 inhibits the line editing function for the current KbdStringIn request.

KbdStringIn completes when the handle has access to the physical keyboard (focus), or is equal to zero and no other handle has the focus.

Family API Considerations

Some options operate differently in the DOS mode than in the OS/2 mode. Therefore, the following restrictions apply to KbdStringIn when coding in the DOS mode: * KbdHandle is ignored

Refer to the DosRead Family API Considerations for differences between DOS and OS/2 node when reading from a handle opened to the CON device.

Example Code

C Binding

<PRE> typedef struct _STRINGINBUF { /* kbsi */

```
USHORT cb;                      /* input buffer length */
USHORT cchIn;                   /* received input length */
```

} STRINGINBUF;

#define INCL_KBD

USHORT rc = KbdStringIn(CharBuffer, Length, IOWait, KbdHandle);

PCH CharBuffer; /* Char string buffer */ PSTRINGINBUF Length; /* Length table */ USHORT IOWait; /* Indicate if wait for char */ HKBD KbdHandle; /* Keyboard handle */

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

MASM Binding

<PRE> STRINGINBUF struc

```
kbsi_cb    dw  ? ;input buffer length
```

```
kbsi_cchIn dw  ? ;received input length
```

STRINGINBUF ends

EXTRN KbdStringIn:FAR INCL_KBD EQU 1

PUSH@ OTHER CharBuffer ;Char string buffer PUSH@ OTHER Length ;Length table PUSH WORD IOWait ;Indicate if wait for char PUSH WORD KbdHandle ;Keyboard handle CALL KbdStringIn

Returns WORD </PRE>

Note

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