

This call creates a new logical keyboard.

### Syntax

KbdOpen (KbdHandle)

### Parameters

;KbdHandle (PHKBD) - output : Address of the logical keyboard.

### Return Code

;rc (USHORT) - return:Return code descriptions are: \* 0 NO\_ERROR \*440  
 ERROR\_KBD\_NO\_MORE\_HANDLE \*441 ERROR\_KBD\_CANNOT\_CREATE\_KCB \*464  
 ERROR\_KBD\_DETACHED \*504 ERROR\_KBD\_EXTENDED\_SG

### Remarks

KbdOpen blocks while another thread has the keyboard focus (by way of [KbdGetFocus](#)) until the thread with the focus issues [KbdFreeFocus](#). Therefore, to prevent KbdOpen from blocking, it is recommended that KbdOpen be issued only while the current thread has the focus. For example:

```
;KbdGetFocus : wait until focus available on handle 0 ;KbdOpen : get a logical keyboard handle
;KbdOpen : get another logical keyboard handle ;KbdOpen : get yet another logical keyboard handle
;KbdFreeFocus : give up the focus on handle 0.
```

### Bindings

#### C

```
<PRE> #define INCL_KBD
USHORT rc = KbdOpen(KbdHandle); PHKBD KbdHandle; /* Keyboard handle */
USHORT rc; /* return code */ </PRE>
```

#### MASM

```
<PRE> EXTRN KbdOpen:FAR INCL_KBD EQU 1
PUSH@ WORD KbdHandle ;Keyboard handle CALL KbdOpen
Returns WORD </PRE>
```

## Kbd

From:

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

Permanent link:

<https://www.osfree.org/doku/doku.php?id=en:docs:fapi:kbdopen&rev=1633521475>

Last update: **2021/10/06 11:57**

