

This call closes the existing logical keyboard identified by the keyboard handle.

### Syntax

KbdClose (KbdHandle)

### Parameters

;KbdHandle ([HKBD](#)) - input: Default keyboard or the logical keyboard.

### Return Code

;rc (USHORT) - return:Return code descriptions are: \*0 - NO\_ERROR \*439 - ERROR\_KBD\_INVALID\_HANDLE \*464 - ERROR\_KBD\_DETACHED \*504 - ERROR\_KBD\_EXTENDED\_SG

### Remarks

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

```
;KbdGetFocus:Wait until focus available on handle 0. ;KbdClose:Close a logical keyboard handle.
;KbdClose:Close another logical keyboard handle. ;KbdClose:Close still 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:

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

Permanent link:

<http://ftp.osfree.org/doku/doku.php?id=en:docs:fapi:kbdclose&rev=1633521528>

Last update: **2021/10/06 11:58**

