




Disabilitare il pulsante di chiusura di un form



http://www.vbsimple.net/howto/ht_010.htm

Difficoltà:  2 / 5

In alcuni programmi si può rendere necessario disabilitare il pulsante di chiusura  di un form. Esistono due soluzioni: la prima consiste nello sfruttare l'evento Unload di un form, presentata in un altro HowTo; la seconda soluzione consiste nel disabilitare il pulsante attraverso alcune funzioni API.

Una volta disabilitato il pulsante esso apparirà ombreggiato, risultando, di fatto, inutilizzabile.



Per disabilitare il pulsante di chiusura rimuoveremo la voce **Chiudi** dal menu di sistema  del form. Prima di iniziare con il codice, inseriamo sopra un form un pulsante  di nome **Chiudi** che effettuerà la chiusura del form al posto del normale pulsante di chiusura.

Passiamo subito alla dichiarazione delle funzioni API:

```
1. Option Explicit
2.
3. Private Declare Function GetSystemMenu Lib "user32" (ByVal hwnd As Long, ByVal
  bRevert As Long) As Long
4. Private Declare Function GetMenuItemCount Lib "user32" (ByVal hMenu As Long) As
  Long
5. Private Declare Function RemoveMenu Lib "user32" (ByVal hMenu As Long, ByVal
  nPosition As Long, ByVal wFlags As Long) As Long
6. Private Declare Function DrawMenuBar Lib "user32" (ByVal hwnd As Long) As Long
7. Private Const MF_BYPOSITION = &H400&
```

Abbiamo dichiarato 4 funzioni [API](#) ed una costante.

La prima funzione è la *GetSystemMenu* che ci restituisce l'[handle](#) della barra dei menu di una finestra.

La seconda funzione API è *GetMenuItemCount* che, partendo da un handle di menu, ci riporta il numero di voci contenute in tale menu.

La terza funzione è *RemoveMenu* che elimina una voce di menu.

L'ultima funzione è la *DrawMenuBar* che ridisegna completamente un menu. Quando vengono apportate modifiche ad un menu è necessario comunicare al programma di ridisegnare la barra dei menu tramite questa funzione.

Alla riga 7 abbiamo dichiarato una costante **MF_BYPOSITION** che servirà da parametro per la funzione *RemoveMenu*.

Nel nostro progetto effettueremo la disabilitazione del menu al caricamento del form, identificato dall'evento *Load* del form. Vediamo le istruzioni:

```
9. Private Sub Form_Load()
10.     Dim HandleMenu As Long
```

```

11.      Dim VociMenu As Long
12.      HandleMenu = GetSystemMenu(Me.hwnd, 0)
13.      VociMenu = GetMenuItemCount(HandleMenu)
14.      Call RemoveMenu(HandleMenu, VociMenu - 1, MF_BYPOSITION)
15.      Call RemoveMenu(HandleMenu, VociMenu - 2, MF_BYPOSITION)
16.      DrawMenuBar Me.hwnd
17. End Sub

```

La riga 10 definisce una variabile di nome **HandleMenu** che utilizzeremo per memorizzare l'[handle](#) del menu di sistema. Come quasi tutte le variabili utilizzate dalle funzioni API, anche questa è dichiarata come Long.

Alla riga 11 dichiariamo un'altra variabile: **VociMenu**. Essa conterrà il numero di voci contenute all'interno del menu di sistema. Fra poco vedremo a che serve.

Come prima operazione otterremo l'handle del menu di sistema del form. La funzione *GetSystemMenu* richiede come parametro l'handle della finestra contenente il menu. Il valore restituito da questa funzione verrà memorizzato all'interno di **HandleMenu**. D'ora in poi utilizzeremo questa variabile per indicare alle altre funzioni API il menu di sistema.

Alla riga 13 utilizzeremo la funzione *GetMenuItemCount* passandogli come parametro l'handle contenuto all'interno della variabile **HandleMenu**. Questa funzione ci restituisce un numero intero rappresentante il numero di voci contenute all'interno del menu, valore che memorizzeremo nella variabile **VociMenu**.

Quest'operazione è necessaria poiché la funzione che rimuove una voce di menu richiede la sua posizione ordinale all'interno del menu. Ebbene, analizziamo queste tre situazioni:

BorderStyle = Sizable MinButton = True MaxButton = True	BorderStyle = Sizable MinButton = True MaxButton = False	BorderStyle = Fixed Single MinButton = False MaxButton = False

Se in fase di progettazione cambiamo alcune proprietà del form, viene alterato il menu di sistema del form. Ragion per cui non possiamo decidere a priori quale elemento eliminare. Però abbiamo una situazione utile dal nostro lato: in tutti i casi la voce **Chiudi** è sempre l'ultima ed il separatore è sempre prima di essa.

Ecco il perché utilizzeremo la funzione *GetMenuItemCount*: essa ci dirà di quanti elementi è composto il menu di sistema. Conoscendo questo provvederemo ad eliminare gli ultimi due elementi.

```

14.      Call RemoveMenu(HandleMenu, VociMenu - 1, MF_BYPOSITION)
15.      Call RemoveMenu(HandleMenu, VociMenu - 2, MF_BYPOSITION)
16.      DrawMenuBar Me.hwnd
17. End Sub

```

Quando si effettua l'eliminazione di più voci di menu è bene tenere a mente due cose: i


menu iniziano il loro conteggio da 0 e subito dopo l'eliminazione di una voce da un menu, il numero complessivo diminuisce di 1. Questo significa che se, inizialmente, le voci di menu sono 7 e vogliamo eliminare le ultime due voci, dobbiamo comunicare all'API di cancellare dal menu le voci 6 e 7, ma poiché le voci sono a base 0, i numeri da passare per eliminare sono 5 e 6. Ma, ricordiamo questa cosa importante: se eliminiamo la voce 5 del menu, tutte le successive diminuiscono di una posizione. L'originaria 6 diventa 5. Questo significa che dopo aver eliminato la voce 5 dovremmo eliminare nuovamente la 5, che in realtà è la 6 precedente all'eliminazione. Per evitare spiacevoli inconvenienti è fondamentale effettuare il conteggio partendo dal numero più alto.

Ecco perché alla riga 14, eliminiamo l'ultima riga (VociMenu - 1) ed in seguito eliminiamo la penultima (VociMenu - 2).

In seguito alla modifica, alla riga 17, comunichiamo all'API di ridisegnare la barra dei menu tramite la funzione *DrawMenuBar*.

```
19. Private Sub Chiudi_Click()  
20.     Unload Me  
21. End Sub
```

Prima di provare il programma è necessario scrivere una funzione sostitutiva al pulsante di chiusura, che abbiamo disabilitato. Al click sopra il pulsante **Chiudi** abbiamo comunicato al form di chiudere se stesso.

Naturalmente si consiglia di non abusare di comandi del genere. Lo standard Windows stabilisce che il metodo più naturale per chiudere un form è il pulsante di chiusura  ed è naturale che un utente, per prima cosa andrà a cercarsi tale pulsante.

Inoltre, una volta disabilitato il pulsante di chiusura, non sarà possibile riabilitarlo.

Prima di chiudere questo argomento voglio rivelarvi un piccolo segreto: esiste un modo più semplice per eliminare la voce **Chiudi** di un form. Abbiamo voluto utilizzare questa soluzione per stimolare il calcolo e permettere l'estensione del codice. Per effettuare la medesima operazione basteranno queste semplici righe:

```
1. Option Explicit  
2.  
3. Private Declare Function GetSystemMenu Lib "user32" (ByVal hwnd As Long, ByVal  
   bRevert As Long) As Long  
4. Private Declare Function RemoveMenu Lib "user32" (ByVal hMenu As Long, ByVal  
   nPosition As Long, ByVal wFlags As Long) As Long  
5. Private Declare Function DrawMenuBar Lib "user32" (ByVal hwnd As Long) As Long  
6. Private Const MF_REMOVE = &H100&  
7. Private Const SC_CLOSE = &HF060&  
8.  
9. Private Sub Form_Load()  
10.     Dim HandleMenu As Long  
11.     HandleMenu = GetSystemMenu(Me.hwnd, 0)  
12.     Call RemoveMenu(HandleMenu, SC_CLOSE, MF_REMOVE)  
13.     DrawMenuBar Me.hwnd  
14. End Sub
```

[Fibia FBI](#)

9 Dicembre 2000



[Torna all'indice degli HowTo](#)

