

[Home Page](#) [Informazioni](#) [Aiuto](#)

Bloccare un form sempre in primo piano

http://www.vbsimple.net/howto/ht_043.htm

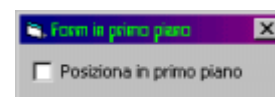
Difficoltà: ► 1 / 5

Dall'avvento di Windows 95 il numero dei programmi utilizzati contemporaneamente si è notevolmente innalzato. È infatti possibile avviare più programmi ed utilizzare quello più utile al momento con un semplice click sulla barra delle applicazioni.

L'attivazione, però di una finestra comporta quasi sempre la sparizione della finestra precedentemente in primo piano. In alcuni casi è però necessario che una determinata finestra rimanga sempre e comunque in primo piano, anche se ne viene attivata un'altra.

Vedremo in questo semplice HowTo come porre una finestra in primo piano rendendola quindi sempre presente nonostante vengano attivate altre finestre. Il trucco sta soltanto nell'uso della funzione [API SetWindowPos](#), in grado di modificare le dimensioni e le posizioni X, Y o Z di una finestra.

Il progetto sviluppato si compone di un solo form con un singolo controllo *CheckBox* di nome **chkPrimoPiano** ed il codice si rivela altrettanto semplice:



```
1. Option Explicit
2.
3. Private Declare Function SetWindowPos Lib "user32" (ByVal hwnd As Long, ByVal
  hwndInsertAfter As Long, ByVal x As Long, ByVal y As Long, ByVal cx As Long, ByVal
  cy As Long, ByVal wFlags As Long) As Long
4.
5. Private Const HWND_TOPMOST = -1
6. Private Const HWND_NOTOPMOST = -2
7. Private Const SWP_NOSIZE = 1
8. Private Const SWP_NOMOVE = 2
9.
10. Private Const lngNoChangePos = SWP_NOMOVE + SWP_NOSIZE
11.
```

Alla riga 3 è dichiarata la funzione *SetWindowPos* che richiede quali argomenti l'[handle](#) della finestra da posizionare, le posizioni Z (**hwndInsertAfter**), X ed Y, le grandezze larghezza ed altezza ed un valore che specifica il tipo di operazioni da effettuare.

Alle righe 5 e 6 sono dichiarati due valori API che specificano il blocco e lo sblocco della posizione Z: **HWND_TOPMOST** specifica che la finestra dovrà sempre e comunque stare in primo piano, anche se disattivata, mentre **HWND_NOTOPMOST** disattiva questo comportamento.

Poiché per i nostri fini non è necessario spostare o ridimensionare la finestra ma la funzione *SetWindowPos* richiede comunque un valore, è possibile utilizzare un valore di flag che specifichi di non utilizzare i valori passati come posizione X ed Y e come dimensioni di larghezza ed altezza. In tal modo la funzione effettuerà soltanto il blocco sull'asse Z. Le costanti API alle righe 7 ed 8 serviranno proprio a questo scopo: non effettuare lo spostamento né il ridimensionamento. Per semplificare la scrittura del codice

abbiamo aggiunto anche una terza costante, non definita nell'API, che contenga gli ultimi due valori da passare alla funzione `SetWindowPos`.

```
12. Private Sub chkPrimoPiano_Click()  
13.     If chkPrimoPiano.Value = vbChecked Then  
14.         Call SetWindowPos(Me.hwnd, HWND_TOPMOST, 0, 0, 0, 0, lngNoChangePos)  
15.     Else  
16.         Call SetWindowPos(Me.hwnd, HWND_NOTOPMOST, 0, 0, 0, 0, lngNoChangePos)  
17.     End If  
18. End Sub
```

Nel momento in cui l'utente clicca sulla casella di controllo ☒ sarà effettuato un controllo del suo valore per permettere il blocco o lo sblocco sull'asse Z.

Se la casella è selezionata (valore ***vbChecked*** - riga 13) sarà richiamata la funzione `SetWindowPos` con l'impostazione ***HWND_TOPMOST*** per bloccare la finestra in primo piano; nel caso inverso (riga 16) sarà richiamata la stessa funzione con il valore ***HWND_NOTOPMOST*** per disattivare il blocco precedentemente attivato.

La costante ***lngNoChangePos*** specifica di non modificare né posizione X ed Y né dimensioni della finestra, ignorando quindi quella serie di 0 passati soltanto perché la funzione richiedeva i parametri.

Il funzionamento del progetto è davvero molto semplice:



Figura 2



Figura 3

Nel momento in cui la casella di controllo non è selezionata le altre finestre (nel nostro caso il programma **APIViewer 2001**) potranno posizionarsi di fronte al nostro form come mostrato nella Figura 2.

Quando la casella di controllo viene selezionata, l'attivazione di un'altra applicazione come dimostra il colore blu della barra del titolo non nasconderà il nostro form, che rimarrà sempre e comunque in primo piano pur non essendo più il form attivo al momento, come mostrato nella Figura 3.

Il progetto è davvero molto molto semplice e si è preferito presentare una scrittura leggermente più lunga ma più esplicativa e non presenta alcun difetto particolare.

Soltanto si raccomanda di non abusare di questa funzionalità di Windows perché in molte situazioni può risultare davvero fastidioso trovarsi una finestra non desiderata dinanzi e non poterla nascondere dietro le altre finestre. L'utilizzo ideale è quello di lasciar decidere all'utente se essa dovrà stare in primo piano mediante un pannello di opzioni ed in ogni caso dovrebbe esistere qualche ragione valida per cui essa debba stare sempre in primo piano.



[Torna all'indice degli HowTo](#)
