Scorrere orizzontalmente una ListBox

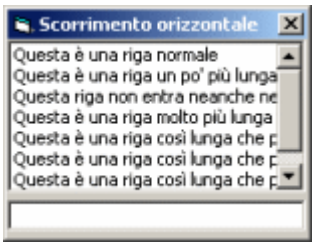
http://www.vbsimple.net/howto/ht_060.htm

Difficoltà: 2 / 5

Quante volte vi sarà capitata lo sgradevole comportamento dei controlli ListBox che presentano soltanto la barra di scorrimento verticale per visualizzare le righe che rientrano nell'area visibile del controllo; manca infatti del tutto la barra di scorrimento orizzontale, ad esempio per scorrere verso destra la visuale e mostrare ciò che va oltre l'area visibile determinata dalla larghezza della ListBox.

Non esiste alcuna proprietà del controllo per abilitare le barre di scorrimento orizzontali ma basta una semplice chiamata ad una funzione dell'API per estendere l'area del controllo e quindi scorrere verso destra il resto del testo. Più precisamente dovremo inviare il messaggio ***LB_SETHORIZONTALEXTENT*** e specificare la larghezza in pixel dell'area da scorrere.

Questa soluzione si presenta così facile che mi si è presentato il dubbio sulla necessità di pubblicare una soluzione al problema.



Per questo progetto di esempio poseremo sopra un form un controllo *ListBox* di nome **lbElenco** ed un controllo *TextBox* di nome **txtTesto** e gestiremo l'evento KeyPress e verificare se è stato premuto Enter. Ciò che stiamo sostanzialmente aggiungendo è la semplice possibilità di inserire dati in una ListBox a scelta dell'utente. Ad ogni inserimento ricalcoleremo la larghezza massima necessaria per visualizzare tutto il testo.

Senza una gestione della larghezza il testo apparirebbe come nella figura mostrata a fianco, mozzato e impossibile da scorrere verso destra. Il codice si compone di due sole routine:

```
 1. Option Explicit
 2.
 3. Private Const LB_SETHORIZONTALEXTENT = &H194
 4.
 5. Private Declare Function SendMessage Lib "USER32" Alias
    "SendMessageA" (ByVal hWnd As Long, ByVal wMsg As Long, ByVal wParam As
    Long, lParam As Any) As Long
 6.
 7. Private Sub txtTesto_KeyPress(KeyAscii As Integer)
 8.     If KeyAscii = vbKeyReturn Then
 9.         lbElenco.AddItem txtTesto.Text
10.         ListBoxAutoScroll lbElenco
11.     End If
12. End Sub
13.
```

Il codice non dovrebbe richiedere alcun commento fin qui; nel caso venisse premuto il pulsante Enter all'interno della casella di testo, sarà inserito il testo della casella all'interno

dell'elenco e verrà quindi richiamata la funzione **ListBoxAutoScroll** che vediamo qui subito esposta:

```
14. Private Sub ListBoxAutoScroll(ByRef LB As ListBox)
15.     Dim lngLarghezza As Long
16.     Dim lngMax As Long
17.     Dim intLoop As Integer
18.     Set Me.Font = LB.Font
19.     For intLoop = 0 To LB.ListCount - 1
20.         lngLarghezza = Me.TextWidth(LB.List(intLoop))
21.         If lngLarghezza > lngMax Then lngMax = lngLarghezza
22.     Next intLoop
23.     lngMax = IIf(lngMax <= LB.Width, 0, lngMax + Me.ScaleX(12, vbPixels,
    vbTwips))
24.     lngMax = Me.ScaleX(lngMax, vbTwips, vbPixels)
25.     Call SendMessage(LB.hWnd, LB_SETHORIZONTALEXTENT, lngMax, ByVal 0&)
26. End Sub
```

La routine **ListBoxAutoScroll** sostanzialmente calcola la larghezza massima necessaria per visualizzare completamente il testo; per far ciò effettua un ciclo su tutti gli elementi della ListBox (righe 19-22) ed ogni volta estrae la larghezza del testo di ogni riga; per far ciò utilizza la funzione *TextWidth*, che fra l'altro per operare correttamente necessita che il carattere utilizzato dal form sia lo stesso di quello usato nel controllo contenente il testo e pertanto vi abbiamo assegnato lo stesso font (riga 18).

Di volta in volta la nuova larghezza, se superiore a quella indicata in **lngMax** prenderà il posto di quest'ultima; in tal modo all'uscita del ciclo avremo all'interno di lngMax la larghezza della riga con il testo più lungo. Tuttavia questo dato è rappresentato nell'unità di misura preferita di Visual Basic, i twips.

Alla riga 23 è verificato che la nuova larghezza sia superiore alla larghezza dell'intera ListBox; se non dovesse esserlo il valore di lngMax verrà azzerato; in caso contrario saranno aggiunti 12 pixel (convertiti da pixel a twips mediante **ScaleX** già trattata in un altro articolo di questa sezione) ed infine l'intero valore di lngMax verrà riconvertito in pixel, l'unità di misura richiesta dalle funzioni dell'API.

Al termine di questa operazione lngMax conterrà il valore 0 oppure la larghezza in pixel del testo più lungo dell'elenco. Questo valore verrà utilizzato per specificare la nuova larghezza virtuale dell'area della ListBox e per far ciò invieremo il messaggio ***LB_SETHORIZONTALEXTENT*** alla ListBox, come indicato alla riga 25.

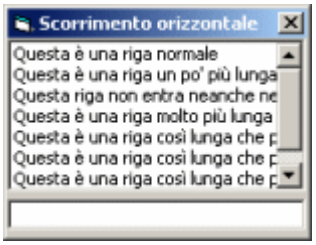


Figura 2

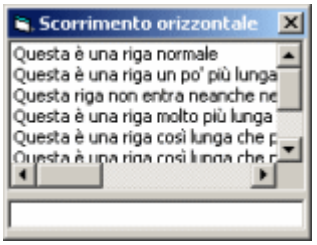


Figura 3

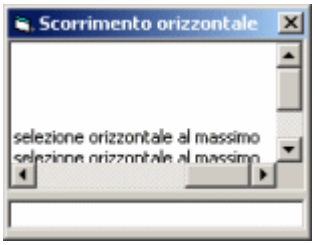


Figura 4

Nella Figura 2 è mostrato nuovamente il testo come apparirebbe se non avessimo usato la nostra funzione **ListBoxAutoScroll**; le figure 3 e 4 mostrano invece la nuova situazione, il etsto inserito supera la larghezza dell'elenco ed in basso appare la barra di scorrimento orizzontale. Scorriamo il testo fino all'estrema destra e possiamo notare come l'area di

scorrimento coincida con la larghezza del testo (più i 12 pixel che abbiamo aggiunto per non far sembrare il tutto allineato al bordo).

Una soluzione facile, veloce e abbastanza sicura per un problema noioso che avrebbe richiesto una proprietà nativa dei controlli ListBox.

Fibia FBI
29 Marzo 2004


Torna all'indice degli HowTo

