

Invio dati ad un'applicazione esterna

http://www.vbsimple.net/activity/act_23_2.htm

Richiesta di: [LoneFlier](#) - 2 Settembre 2002

Difficoltà: 3 / 5

Ho scritto un programma in VB [...] non riesco però a fare in modo che, dopo essere ridotto ad icona, mi invii i dati all'applicazione attiva, che non dovrebbe essere definita dal programma.

[<< Prima Parte](#)

Nella prima parte di quest'articolo abbiamo presentato una lunga e complessa soluzione in cui il 90% delle spiegazioni erano incentrate sul recupero della finestra che dovrà ricevere i dati da inviare. In questa seconda parte invece sarà proposta una soluzione molto più semplice, compatta e probabilmente migliore.

La soluzione si basa fondamentalmente nell'uso di un controllo [FBI Select Window](#) al quale saranno delegate tutte le operazioni di ricerca della finestra di destinazione. Basterà un semplice click ed un trascinamento per decidere dove inviare i dati.

L'interfaccia del nostro progetto cambierà davvero di poco: sarà aggiunto un controllo *FBI Select Window* e verrà lasciato il nome standard **FBISelectWindow1**.



Anche il codice sarà molto simile al precedente ma semplicemente più snello:

```
1. Option Explicit
2.
3. Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
4. Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hWnd
   As Long, ByVal wParam As Long, ByVal lParam As Any) As Long
5.
6. Private Const WM_CHAR = &H102
7. Private Const WM_SETTEXT = &HC
8. Private Const EM_SETSEL = &HB1
9.
```

Tutte le funzioni di intercettazione della finestra attiva sono state rimosse e sono rimaste solo le due funzioni fondamentali: **SendMessage** per inviare un [messaggio](#) alla [Window Procedure](#) di una finestra e **Sleep** per [effettuare una pausa](#). I messaggi da inviare sono sempre gli stessi: **WM_CHAR**, **WM_SETTEXT** e **EM_SETSEL** per una cui spiegazione si rimanda alla Prima Parte dell'articolo.

```
10. Private Sub Form_Load()
11.     Call Shell("NOTEPAD.EXE", vbNormalNoFocus)
12. End Sub
13.
```

Anche la routine all'avvio del Form è rimasta pressoché identica. Sarà infatti caricato il Blocco Note di Windows e messo in secondo piano.

```

14. Private Sub cmdInvia_Click()
15.     Dim lngHwnd As Long
16.     Dim intConta As Integer
17.     Dim bytBuffer() As Byte
18.     Const lngTypeDelay = 0
19.     lngHwnd = FBISelectWindow1.LastWindow
20.     bytBuffer = StrConv(txtBla.Text & vbCrLf & vbNullChar, vbFromUnicode)
21.     Call SendMessage(lngHwnd, WM_SETTEXT, ByVal 0&, bytBuffer(0))
22.     Call SendMessage(lngHwnd, EM_SETSEL, UBound(bytBuffer, 1), 0)
23.     For intConta = 0 To UBound(bytBuffer, 1) - 2
24.         Call SendMessage(lngHwnd, WM_CHAR, bytBuffer(intConta), ByVal 0&)
25.         If lngTypeDelay > 0 Then Call Sleep(lngTypeDelay)
26.     Next intConta
27. End Sub

```

La routine che gestisce il click del pulsante **cmdInvia** è diventata un attimo più semplice e snella: le uniche variabili rimaste sono quelle che indicano l'[handle](#) della finestra client a cui inviare i dati, la variabile utilizzata per il ciclo e la [matrice](#) di bytes che servirà come [buffer](#) dei dati da inviare.

Alla riga 19 è recuperato l'handle dell'ultima finestra selezionata dall'utente; alla riga successiva la stringa della casella di testo è convertita in matrice di bytes con le [convenzioni delle stringhe API](#) (cioè aggiungendo il carattere [NULL](#) alla fine).

In maniera identica all'articolo precedente, sarà adesso possibile inviare i dati alla finestra client di destinazione. L'invio anche in questo caso è effettuato in due maniere: sostituendo il testo precedente (messaggio **WM_SETTEXT**) oppure inviando un carattere per volta (messaggio **WM_CHAR**). Per una spiegazione più accurata si rimanda alla prima parte dell'articolo. 😊

La dimostrazione dell'applicazione è altrettanto semplice: avviato il progetto, sarà compito dell'utente trascinare il mirino sopra la finestra dentro la quale si desidera che vengano inviati i dati. Se la proprietà **InvertOnSelect** del controllo FBI Select Window è impostata su True, la selezione della finestra puntata è evidenziata dall'inversione dei colori.

Unico requisito per l'utente: la cura nel puntare la finestra [Child](#) e non quella [Parent](#) dell'applicazione. La differenza tra le due finestre è mostrata nelle Figura 2 e 3.

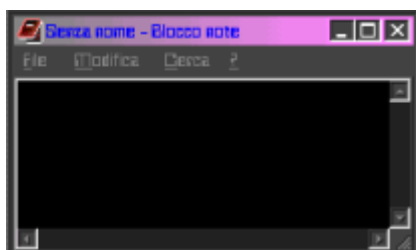


Figura 2
(Scelta la finestra Parent)

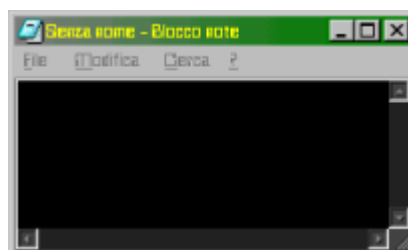


Figura 3
(Scelta la finestra Child)

La Figura 2 mostra la selezione della finestra Parent, ad esempio posando il mirino sulla barra del titolo oppure sulla barra dei menu. L'utilizzo di questa finestra invierebbe il testo sulla barra del titolo dell'applicazione Blocco Note, al posto di **"Senza nome - Blocco**

note''.

La Figura 3 mostra invece la maniera d'uso corretta per questo esempio: è stata scelta la finestra Child, la grande casella di testo (che nella figura appare però nera). L'invio dei dati a quella finestra produrrà l'effetto voluto, cioè quello di scrivere del testo all'interno dell'applicazione Blocco Note.

Questa soluzione presenta numerosi indiscussi vantaggi: il primo riguarda la semplicità d'uso, non obbligando l'utente ad un movimento rapido e preciso per la selezione della finestra.

Inoltre, una volta scelta la finestra, fino a quando questa non verrà chiusa, la selezione è valida. L'utente può in ogni momento cambiare finestra attiva, spostarla, ridurla ad icona e quant'altro, senza alcuna fretta ed il programma continuerà a puntare la finestra corretta scelta in precedenza. E se l'utente dovesse ripensarci, ha tutto il tempo di scegliere una nuova finestra.

Il controllo FBI Select Window per questo genere di operazioni semplifica notevolmente le incombenze del programmatore. In questa maniera però è obbligatorio un intervento dell'utente, per selezionare la finestra che si intende utilizzare.

In un progetto più serio e complesso che utilizza una tecnica del genere si raccomanda qualche informazione in più per l'utente circa la finestra puntata, quale l'handle, la classe, il [processo](#) di appartenenza o la sua Caption. Questo informerà l'utente che la sua selezione è corretta non solo visivamente ma anche a livello di sistema. Si raccomanda la visione del progetto sviluppato nell'[articolo dedicato al controllo FBI Select Window](#).

[Fibia FBI](#)

14 Ottobre 2002



[Torna all'introduzione delle Richieste dei lettori](#)
