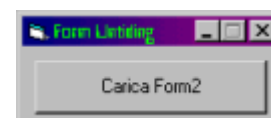


Difficoltà: ► 1 / 5

Nei progetti che utilizzano più form un errore molto comune è quello di trovare riferimenti al secondo form senza un esplicito caricamento. Questo quindi provoca il caricamento automatico del form a cui si fa riferimento; tuttavia questa maniera di lavorare può portare ad errori involontari che, inaspettatamente, lasciano il form in memoria ma in maniera invisibile.

Il significato letterale del termine Forms Untiding significa disordinamento dei forms e si applica in genere su quei forms secondari che al momento dell'avvio sono nascosti, ad esempio per inizializzarne i dati dall'esterno.

Sviluppiamo un brevissimo esempio esplicativo composto da due forms: il primo conterrà un solo pulsante di nome **cmdLoadForm2** mentre il secondo form non necessita di alcun controllo.



Il codice del pulsante fa uso esplicito di alcune proprietà del secondo form, senza tuttavia utilizzare un'istruzione Load che avrebbe piuttosto evidenziato meglio l'errore.

```
1. Option Explicit
2.
3. Private Sub cmdLoadForm2_Click()
4.     Form2.Visible = False
5.     Form2.Caption = "Form nascosto"
6.     If MsgBox("Faccio apparire il form?", vbYesNo) = vbYes Then Form2.Show
7. End Sub
```

Le righe 4 e 5 modificano due proprietà del secondo form forzando il caricamento automatico dello stesso. La riga 6 invece pone una domanda all'utente: **"Faccio apparire il form?"**. Naturalmente al suo posto avrebbe potuto esserci qualunque istruzione di controllo il cui risultato decidesse se visualizzare o meno il secondo form.

Se alla domanda l'utente risponde con SI, il secondo form è mostrato sullo schermo tramite *Form2.Show* e l'elaborazione procede regolarmente. Sarà quindi compito del secondo form scaricarsi quando non serve più oppure lasciare all'utente la possibilità di chiuderlo tramite il pulsante di chiusura sulla barra del titolo.

Se invece l'utente risponde con un NO, il secondo form non sarà mostrato e la routine terminerà normalmente, non avendo altro codice da eseguire. L'utente neanche si accorge del problema in quanto il secondo form è nascosto (e non è stato mostrato); tuttavia questo genera una situazione di *Forms Untiding*. Uno o più forms non sono stati scaricati dalla memoria, complice anche l'assenza di un'esplicita istruzione **Load** che avrebbe ricordato o fatto risaltare l'eventuale necessità di un'istruzione **Unload**.

In una situazione di Forms Untiding se l'utente chiude il form principale, nonché l'unico

visibile, il processo non termina ma rimane sospeso. Se il progetto è eseguito all'interno dell'[IDE](#) di VB ci si accorge del problema per il fatto che l'esecuzione non ritorna alla fase di progettazione; il codice del progetto è visibile ma non modificabile e l'esecuzione risulta ancora in atto.

Se invece il progetto fosse stato compilato ed avviato dal file eseguibile risulta impossibile accorgersi della situazione sospesa. Nascosto ogni form il progetto sembra non esistere e non presenta alcuna icona sulla barra delle applicazioni; ci si può accorgere del processo sospeso soltanto premendo ad esempio i tasti **CTRL+ALT+CANC**.

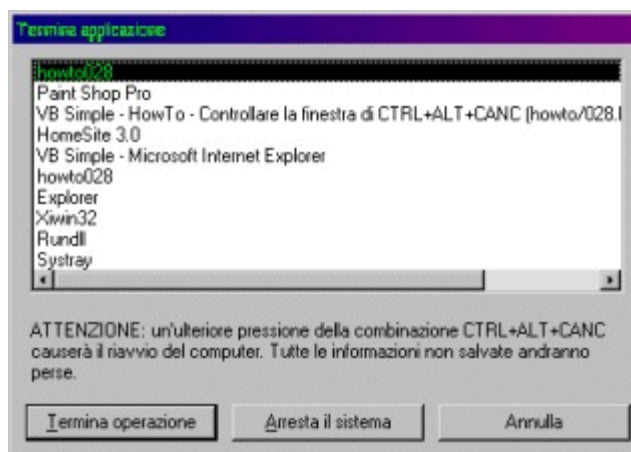


Figura 2

Risolvere una situazione di Forms Untiding è relativamente semplice. Basta applicare un po' di attenzione durante l'utilizzo di form secondari:

- Evitare il caricamento di un form prima dell'effettiva necessità
- Esplicitare le istruzioni Load / Unload
- Verificare i forms caricati in memoria
- Chiudere tutti i forms durante la chiusura del form principale che verrà chiuso per ultimo

Lo stesso progetto poteva essere scritto nel seguente modo:

```
1. Option Explicit
2.
3. Private Sub cmdLoadForm2_Click()
4.     If MsgBox("Faccio apparire il form?", vbYesNo) = vbYes Then
5.         Load Form2
6.         Form2.Visible = False
7.         Form2.Caption = "Form nascosto"
8.         Form2.Show
9.     End If
10. End Sub
11.
```

Questa operazione applica le prime due regole precedenti: il caricamento del form è eseguito soltanto quando è necessario farlo (l'utente ha risposto di voler mostrare il form) ed è esplicitata l'istruzione Load. Se il form fosse stato visualizzato in maniera [modale](#) sarebbe stato necessario aggiungere un'istruzione Unload subito dopo la visualizzazione.

Alla riga 4 viene verificata la condizione che determina se mostrare o meno il secondo form. Soltanto se l'utente decide di voler mostrare il form questo sarà caricato e mostrato. Non così avviene se l'utente decide di non mostrare il form.

Inoltre la chiusura del form principale deve essere eseguita come segue:

```
12. Private Sub Form_Unload(Cancel As Integer)
13.     Dim intConta As Integer
14.     If Forms.Count > 1 Then
15.         For intConta = Forms.Count - 1 To 1 Step -1
16.             Unload Forms(intConta)
17.         Next intConta
18.     End If
19. End Sub
```

Sarà verificato il numero di forms caricati e scaricati i forms non necessari.

Alla chiusura del form principale sarà verificato il numero di forms caricati in memoria. Se il numero di forms è superiore ad 1, ovvero esiste almeno un altro form oltre il principale, saranno chiusi uno per uno i forms secondari iterando ciascun form dall'ultimo al secondo. Il form principale si chiuderà da sé al termine della routine Form_Unload.

Questa è una possibile soluzione. In certi casi anche questo controllo può fallire, ad esempio quando uno dei forms secondari rifiuta la chiusura nell'[evento](#) ⚡ Unload.

[Fibia FBI](#)

23 Ottobre 2002



[Torna all'indice delle Stranezze](#)
