



I Controlli in una Userform

Un oggetto Userform può contenere controlli come possiamo vedere nelle finestre di dialogo normalmente mostrate da Excel o altri programmi. I controlli sono gli elementi di una finestra che consentono all'utente di interagire con il programma. Nella lezione precedente abbiamo fatto una panoramica su questo argomento, adesso vediamo di approfondire elencando i vari controlli disponibili in maniera più dettagliata.

I controlli disponibili in VBA sono:

Label : Corrisponde a Etichetta, inserisce un oggetto di tipo label1 e svolge la funzione di etichetta descrittiva per quei controlli che non ne hanno una propria. E' inoltre possibile utilizzare le etichette per mostrare il valore di una variabile. Si può modificare il testo mostrato di una etichetta (label1 nel nostro caso) modificando la proprietà Caption, questo è possibile via codice oppure agendo nella finestra delle proprietà, generalmente questo oggetto viene usato nei casi in cui non si debba modificarne il testo, ma semplicemente come elemento descrittivo di un altro controllo

Casella di testo : Inserisce un oggetto di tipo Textbox, questo oggetto permette l'inserimento di testo da parte dell'utente, viene utilizzato per ogni tipo di campo che può essere rappresentato come testo, nomi, date, numeri valute e inoltre può accettare e mostrare più righe di testo, infatti quando necessario compare una barra di scorrimento verticale che permette di scorrere il testo contenuto

Cornice : Inserisce un oggetto di tipo Frame, questo oggetto non ha nessuna funzione specifica, se non quella di raggruppare logicamente un insieme di controlli. Si utilizza solitamente per raggruppare pulsanti di opzione, caselle di controllo e pulsanti interruttore. Usate questo controllo quando volete far risaltare all'utente che i controlli contenuti nella cornice sono in qualche modo collegati tra loro

Pulsante di comando : Inserisce un oggetto di tipo CommandButton, il classico pulsante di standard di Windows che si attiva quando l'utente fa clic su di esso. L'azione svolta al clic sul controllo dipende dal codice che viene messo nell'evento, può compiere operazioni di chiusura (*Close*) di nascondimento (*Hide*), di aggiornamento dei dati, di verifica e così via

Casella di Controllo : Inserisce un oggetto di tipo CheckBox, è composto da un quadrato che contiene un segno di spunta nel caso in cui sia selezionato e da un'etichetta di descrizione. Utilizzate questo controllo per le opzioni che possono avere un valore di tipo Vero o Falso. Può restituire valori True (Vero), quando la casella è selezionata o False (Falso), se la casella non è selezionata, inoltre la presenza di più caselle di controllo non si escludono a vicenda

Pulsante di opzione : Inserisce un oggetto di tipo OptionButton, è composto da un pulsante tondo che contiene una pallina nera nel caso sia selezionato e da un'etichetta di descrizione. Viene usato con una serie di pulsanti di opzione per consentire all'utente una selezione di elementi che si escludono a vicenda

Pulsante Interruttore : Inserisce un oggetto di tipo ToggleButton, mostra lo stato del pulsante Vero o Falso, Acceso o Spento, allo stesso modo di una casella di controllo solo che ha l'aspetto di un pulsante in posizione "Su" o "Giù"

Casella di riepilogo : Inserisce un oggetto di tipo ListBox e mostra una serie di dati presenti nel foglio in cui l'utente può fare una scelta, inoltre la proprietà MultiSelect controlla la possibilità di effettuare la scelta di un solo valore o di più elementi della lista



Casella combinata : Inserisce un oggetto di tipo ComboBox, questo controllo è la combinazione grafica di una casella di testo (TextBox) e di una casella di riepilogo (ListBox) la casella di testo permette la digitazione di dati con la possibilità di suggerire una serie di valori tra cui scegliere attraverso la lista a discesa. Tramite questo controllo è possibile consentire all'utente di inserire un valore non presente nella lista, oppure costringerlo a scegliere un valore tra quelli presenti nella lista. Per poter effettuare questa impostazione bisogna modificare la proprietà *Style* nella finestra delle proprietà

Schede : Inserisce un oggetto di tipo TabStrip, questo controllo consiste in una singola area in cui è possibile inserire altri controlli, è costituito da una serie (modificabile a piacere) di pulsanti di tabulazione. Questo controllo è simile all'oggetto cornice, in quanto l'area non cambia, ma tramite i pulsanti di tabulazione è possibile mostrare dati di diverse categorie. Per esempio in un'anagrafica di un cliente è possibile avere un tabulatore in cui compaiono i dati generali (indirizzo, telefono etc.) un altro tabulatore mostrerà i dati bancari e così via

Pagine : Inserisce un oggetto di tipo MultiPage, questo controllo è simile al controllo Schede (TabStrip) inoltre ogni pagina di questo controllo può contenere diversi controlli distinti. E' da utilizzare quando ogni pagina deve avere un contenuto differente

Barra di scorrimento : Inserisce un oggetto di tipo ScrollBar e permette di scorrere i valori contenuti in una finestra

Casella di selezione : Inserisce un oggetto di tipo SpinButton, è una particolare categoria di casella di testo che permette l'inserimento facilitato e la modifica di dati compresi in un certo intervallo. Viene utilizzato per l'inserimento di valori numerici, date o valori in sequenza e si utilizza insieme ad un controllo etichetta(Label) o casella di testo (TextBox). In pratica facendo clic sulla freccia "su" o "giù" il valore viene incrementato o diminuito

Immagine : Inserisce un oggetto di tipo Image, e permette di inserire un'immagine all'interno della finestra in uno dei seguenti formati *.bmp, *.cur, *.jpeg, *.gif, *.ico

RefEdit : Inserisce un oggetto di tipo RefEdit, questa è una speciale casella di testo che consente di inserire e selezionare intervalli sui fogli di lavoro di Excel

Ogni controllo è un oggetto con proprietà, metodi ed eventi specifici esattamente come le finestre (Userform) che li contiene, è possibile impostare le proprietà dei controlli via codice oppure utilizzando la finestra Proprietà dell'editor di Visual Basic. Una volta aggiunto un controllo alla finestra, potete effettuare una serie di operazioni come: copiare l'oggetto, ridimensionarlo, spostarlo, cancellarlo, modificarne la formattazione, il tipo di carattere, il colore e modificare tutte le sue proprietà. Avrete sicuramente la necessità di fare queste operazioni nel momento in cui perfezionerete la vostra tecnica di programmazione delle finestre di dialogo. Per esempio potete decidere che una casella di testo è troppo ampia o troppo stretta per il testo a cui è destinata, così ne modificherete le dimensioni. Oppure un pulsante di comando con la scritta CommandButton7 avrebbe un significato a dir poco "oscuro" per l'utente che userà la vostra finestra di dialogo.

Per poter inserire un controllo nella nostra Userform dovete fare clic sul pulsante *della Casella degli strumenti* corrispondente al controllo che volete aggiungere nella finestra, in questo modo il puntatore del mouse si trasforma in una croce sottile quando viene posizionato sulla Userform. Posizionate il puntatore a croce sulla finestra nel punto in cui volete inserire il controllo, tenete presente che il punto che scegliete corrisponderà all'angolo superiore sinistro. Fate clic e tenete premuto il pulsante sinistro del mouse, Trascinate ora a destra e in basso fino a che il controllo non raggiunge le dimensioni desiderate, quindi rilasciate il pulsante, a questo punto l'editor di Visual basic inserisce il controllo e il puntatore del mouse ritorna ad essere la freccia standard.



Se proviamo ad inserire una casella di testo (TextBox), con le istruzioni sopra riportate ci comparirà una finestra come quella sotto riportata

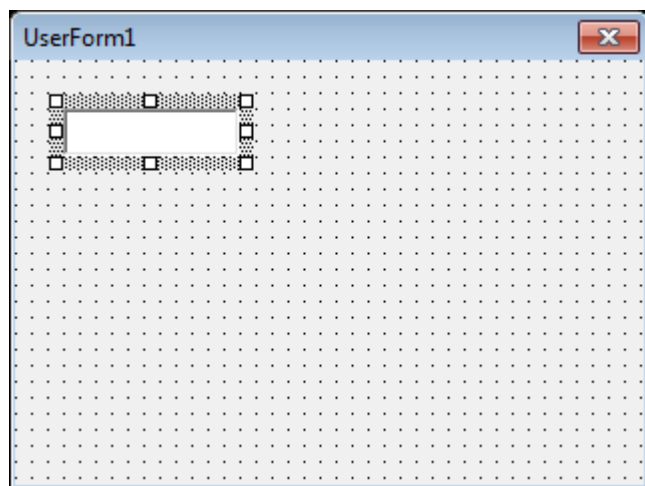


Fig. 1

Avrete notato che il controllo è circoscritto da dei piccoli quadratini bianchi che ci permettono di ridimensionare il controllo stesso, posizionandoci col cursore del mouse su uno di essi il puntatore si trasforma in una doppia freccia, che indica la direzione nella quale è possibile ridimensionare l'oggetto. Inoltre posizionandoci sopra al controllo il cursore cambia aspetto e diventa a 4 frecce, in questa condizione è possibile spostare il controllo nella posizione voluta. Ricordate anche che ogni controllo aggiunto alla finestra di dialogo deve avere un nome univoco, a questo pensa già l'editor di VBA, se inseriamo un controllo `CommandButtons` gli viene assegnato automaticamente il nome del controllo seguito da un numero [es. `CommandButtons1`] questo numero progressivo viene incrementato automaticamente dall'editor ogni volta che si inserisce un controllo dello stesso tipo.

■ **Modificare i controlli**

Una volta che avete aggiunto un controllo alla finestra potete effettuare una serie di operazioni come: copiare l'oggetto, ridimensionarlo, spostarlo, cancellarlo, modificarne la formattazione etc. Avrete sicuramente la necessità di compiere queste operazioni nel momento in cui perfezionerete la vostra tecnica di progettazione delle finestre di dialogo. Per esempio potete decidere che una casella di testo è troppo grande o troppo stretta per il testo che è destinata a ricevere, così necessita di una modifica delle dimensioni, oppure inserire un controllo `CommandButon5` avrebbe un significato oscuro per l'utente che userà il programma.

Per modificare un controllo occorre prima selezionarlo facendo clic sopra al controllo stesso e per modificarlo dobbiamo ricorrere alla finestra delle proprietà. Inseriamo ora per esempio 3 controlli, un `CommandButton`, una `TextBox` e una `Label` nella nostra form che adesso si presenta così

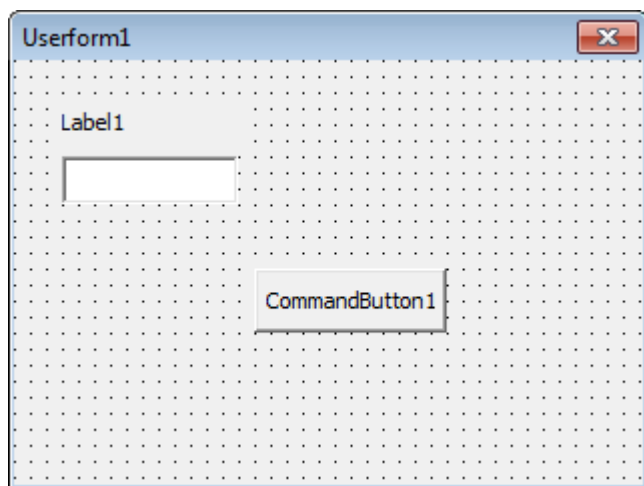


Fig. 2

Presentare una Userform in questo modo non è certamente accattivante, dobbiamo cercare di dare una indicazione visiva all'utente sullo scopo della Userform. Iniziamo modificando i 3 controlli inseriti, trascinandoli nella posizione giusta e utilizzando la finestra delle proprietà per modificare il valore *caption*, modifichiamo anche il titolo della form inserendo la dicitura "Esempio di inserimento", come etichetta della Label usiamo "Inserisci un valore numerico" e il pulsante di comando diventa "Verifica". Come già detto tutte queste modifiche le effettuiamo selezionando il controllo e andando a modificare il valore del campo Caption, come evidenziato dalla *freccia rossa in figura 3*

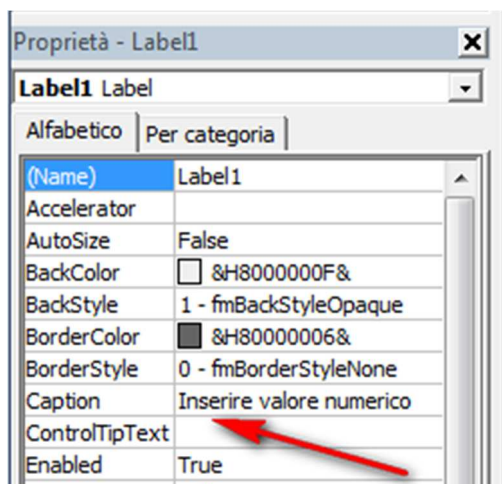


Fig. 3

Una volta terminata la modifica dei vari controlli e ridimensionata la finestra, otteniamo una form come la seguente

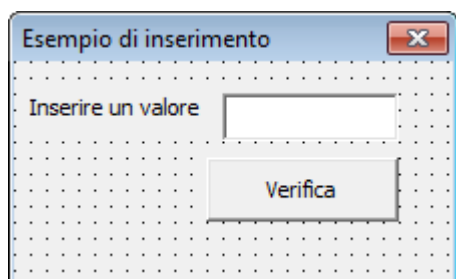


Fig. 4



Possiamo ipotizzare che questa finestra attenda l'inserimento di un valore nella TextBox e al tempo stesso verifichi che sia stato inserito un valore numerico prima di salvarlo in una cella del foglio di lavoro. In questi casi è sempre opportuno fare la verifica del testo inserito per evitare ulteriori errori di runtime nel proseguo del programma. E' possibile fare verifiche del genere in due modi, uno sfruttando l'evento **Change** della TextBox e l'altro con opportuno codice quando clicchiamo sul pulsante di verifica prima di eseguire il salvataggio. Vediamo entrambi i metodi, prima però aggiungiamo un'altra Label alla nostra form che ci torna utile per rimandare un avviso in caso di errore. La possiamo inserire in questo modo

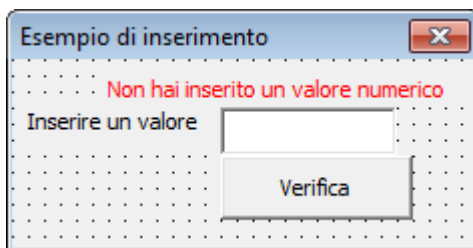


Fig. 5

Come potete vedere è stato cambiato anche il colore del testo agendo sul campo **ForeColor** della finestra delle proprietà come si vede in figura

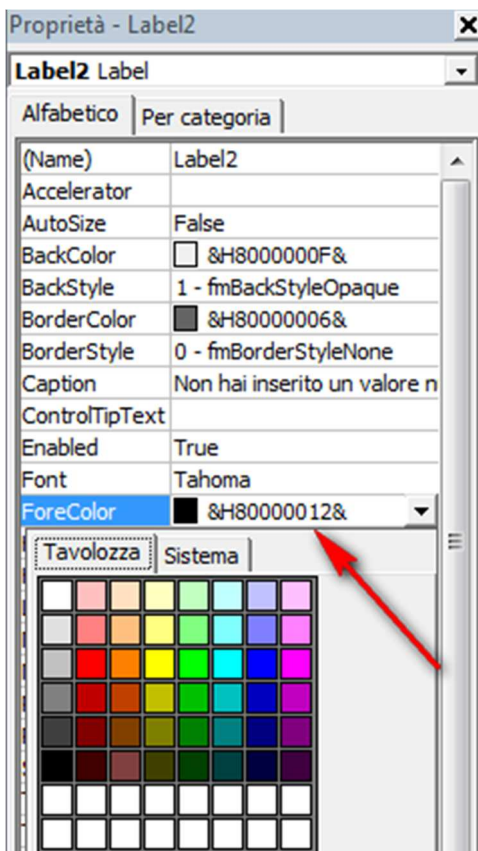


Fig. 6

A questo punto dobbiamo inserire il codice adatto nell'evento Change e per raggiungere questo evento facciamo doppio clic sulla TextBox e nella finestra del codice ci compare l'evento TextBox1_Change(). Abbiamo già detto che questo evento si scatena quando sente un cambiamento all'interno del campo stesso, per cui se inseriamo un codice come il seguente



```
Private Sub Textbox1_Change()  
    If IsNumeric(Textbox1.Value) Then  
        Label2.Visible = False  
    Else  
        Label2.Visible = True  
        TextBox1.Value=""  
    End If  
End Sub
```

Se inseriamo un valore numerico la Label2 **NON** diventa visibile, mentre in caso di valore **NON** numerico la Label2 diventa visibile e la routine termina cancellando il valore nella TextBox. Se eseguiamo questa routine ci viene riportata la form in questo modo

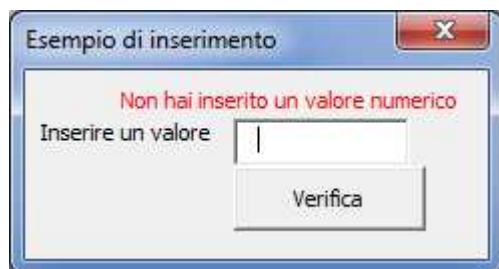


Fig. 7

Come potete vedere la Label2 è visibile, mentre noi vogliamo che diventi visibile in caso di errore nell'inserimento del tipo di dato richiesto. La Label2 è visibile in quanto non abbiamo inizializzato la form, nella lezione precedente abbiamo visto gli eventi **Initialize** e **Activate** delle Userform, nel nostro caso utilizziamo l'evento Initialize per nascondere la Label2 e farla poi apparire in caso di errore tramite il ciclo If presente nell'evento **Change** della TextBox, pertanto andremo ad aggiungere questo codice

```
Private Sub UserForm_Initialize()  
    Label2.Visible = False  
End Sub
```

In questo modo la finestra di dialogo ci compare in questo modo



Fig. 8

A questo punto se inseriamo un valore di tipo stringa nel campo della TextBox ci viene mostrata la Label di avviso errore di inserimento

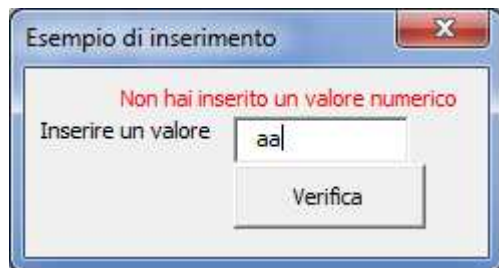


Fig. 9

Per mostrare la Label2 e il testo inserito nel campo della TextBox è stata remmata (è stata resa come un commento e il VBA salta i commenti) un'istruzione nella routine

```
Private Sub Textbox1_Change()  
    If IsNumeric(TextBox1.Value) Then  
        Label2.Visible = False  
    Else  
        Label2.Visible = True  
    ' TextBox1.Value = ""  
    End If  
End Sub
```

Come vedete è stato posto un apostrofo prima dell'istruzione `TextBox1.Value = ""` che è quella che si occupa di svuotare il campo. A questo punto abbiamo visto il primo metodo di verifica dei dati, l'altro, come abbiamo precedentemente accennato, viene eseguito premendo il pulsante di "Verifica" prima di eseguire la scrittura del valore della TextBox nel foglio di lavoro. Lo possiamo fare in questo modo

```
Private Sub CommandButton1_Click()  
    If IsNumeric(Textbox1.Value) Then  
        Range("A1") = Textbox1.Value  
        Unload Me  
    Else  
        MsgBox "Il Valore inserito è errato"  
    End If  
End Sub
```

Se eseguiamo questo codice (ovviamente dobbiamo togliere il codice nell'evento `TextBox1_Change` altrimenti scatta anche quello) in caso di inserimento di un valore stringa ci rimanda questo avviso



Fig. 10