



Introduzione alle Userform con VBA

Finora abbiamo visto come utilizzare delle finestre di dialogo che VBA mette a disposizione tramite le funzioni *MsgBox* e *InputBox*, sebbene queste funzioni possano garantire ad un programma un'ottima funzionalità il loro utilizzo è abbastanza limitato. Durante lo sviluppo di programmi più complessi è indispensabile usare finestre di dialogo, che permettano all'utente di selezionare più opzioni, scegliere elementi da una lista o digitare valori diversi, in pratica per incrementare la funzionalità del programma è necessario utilizzare finestre di dialogo personalizzate che ci permettano di velocizzare le nostre procedure, in questo contesto VBA ci mette a disposizione l'oggetto **Userform** (Finestre Utente) che ci permette la creazione e la manipolazione di finestre di dialogo personalizzate all'interno di programmi o procedure.

Utilizzando le *Userform*, è possibile costruire finestre personalizzate per visualizzare dati, o richiedere all'utente la digitazione di valori, utilizzando la logica che abbiamo impostato per la corretta esecuzione del programma, per esempio possiamo mostrare una finestra di dialogo, che mette a disposizione una serie di formati data, obbligando l'utente alla scelta di un solo formato tra quelli mostrati.

In sostanza le finestre di dialogo permettono al programma di interagire con l'utente in modo più "sophisticato" e forniscono uno strumento versatile per svolgere le normali funzioni di *Input* e *Output*. L'oggetto Userform è una finestra di dialogo vuota e contiene una barra del titolo e un pulsante di chiusura, aggiungendo *controlli* a un oggetto di tipo Userform è possibile personalizzare l'aspetto e la funzionalità della finestra di dialogo. Ogni oggetto Userform possiede *proprietà*, *metodi* e risponde ad *eventi*, tutti ereditati dall'oggetto Userform, inoltre ogni oggetto Userform incorpora un modulo nel quale l'utente può aggiungere i propri metodi e proprietà e nel quale può scrivere il codice che risponde ad eventi della finestra.

Cosa significa questo? Significa che possiamo definire evento qualsiasi cosa si verifichi all'interno della finestra di dialogo o in un suo controllo, tipici esempi di evento sono la pressione di un pulsante di comando o la selezione di una casella di controllo. Altri eventi possono includere la modifica di una casella di testo o la selezione di una lista, i clic del mouse, la pressione dei tasti e altre azioni interne attivano gli eventi. Gli oggetti utilizzati (finestre e controlli) rendono disponibile una serie di eventi, è quindi possibile scrivere procedure VBA che rispondono a questi eventi. Queste procedure vengono denominate "**Procedure di evento**" come per esempio la pressione di un pulsante di comando, la procedura di evento contiene tutte (e solo) le azioni da eseguire in relazione all'evento, altro esempio può essere la chiusura di una finestra tramite il pulsante di chiusura la procedura di evento viene eseguita in aggiunta all'azione causata dall'evento (in questo caso la chiusura della finestra di dialogo)

Come creare una UserForm

Per inserire in un file di Excel una Userform la procedura è abbastanza semplice, entriamo nell'editor del VBA (premiamo i tasti ALT+F11) e nella finestra del progetto vediamo il file xls con l'elenco dei fogli presenti al suo interno. Per creare una Userform seguiamo questo percorso **Inserisci - Userform** e di seguito ci compare una finestra come questa

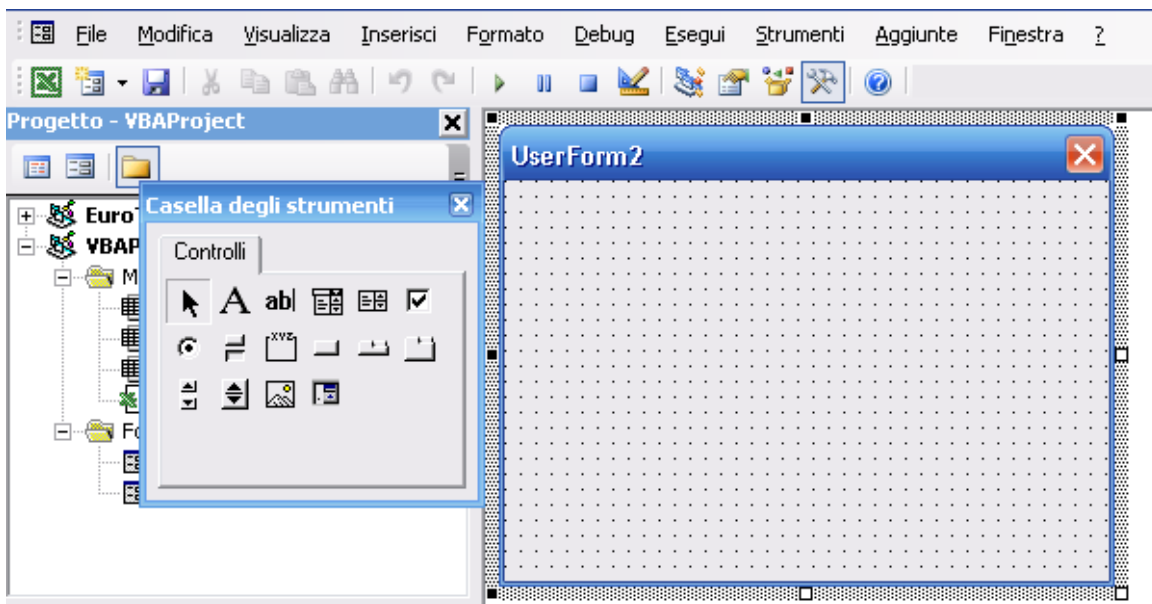


Fig. 1

Nella finestra del codice è comparsa una *UserForm* vuota e a sinistra vediamo un box denominato **Casella degli strumenti** che ci permette di inserire i vari *controlli* all'interno della Userform. La Userform ora è creata, possiamo modificarne le dimensioni a piacere, basta posizionarsi in un angolo (inferiore destro) e trascinare il mouse tenendo premuto il tasto sinistro e rilasciarlo quando abbiamo raggiunto le dimensioni desiderate.

VBA per default assegna il nome Userform seguito da un numero di indice, ma possiamo modificarne il nome anche per sapere a prima vista il compito assegnato alla Userform. Dalla finestra delle proprietà modifichiamo il nome in *Anagrafica*, facendo clic nella casella a fianco del campo (Name), cancelliamo il nome attualmente presente (Userform1) e inseriamo il nuovo.

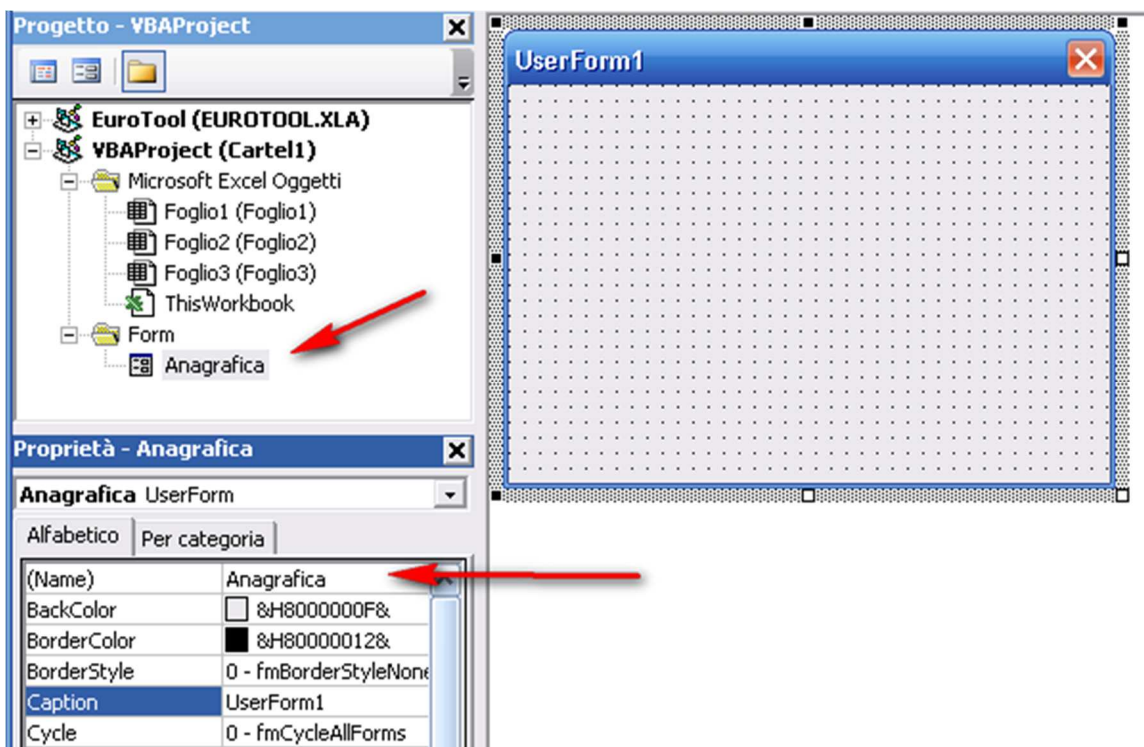


Fig. 2



Ora vediamo che nella finestra dei progetti la UserForm ha cambiato nome, ma il titolo della UserForm è rimasto inalterato, presenta ancora il nome Userform1. Per modificare il titolo sulla barra della Userform dobbiamo modificare il valore presente nel campo *caption* nella finestra delle proprietà, se modifichiamo il nome in Gestione anagrafica anche la barra della Userform assume questo titolo

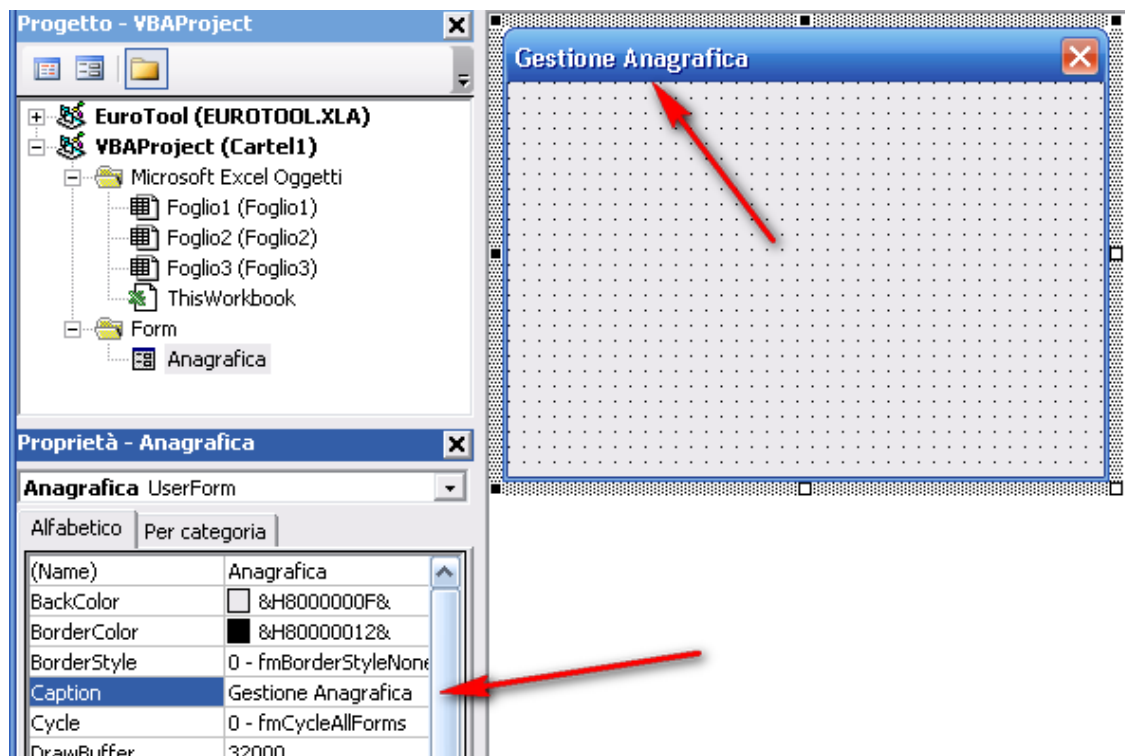


Fig. 3

Una Userform ha i suoi *eventi*, proprio come una cartella di lavoro o un foglio di lavoro ed essendo diversi, per il momento ci fermeremo a quelli più usati che sono: *Activate* – *Click* – *DbClick* – *Deactivate* – *Initialize* – *Terminate*

Per aggiungere eventi, dobbiamo fare doppio clic sulla finestra UserForm:



Fig. 4

Possiamo usare un evento per fissare le dimensioni della Userform, utilizzando l'evento *Userform_Initialize* che scatterà quando la form viene caricata in memoria. Questo evento può essere causato dall'istruzione *Load* o dal metodo *Show*, si utilizza questo evento per impostare l'aspetto iniziale della finestra. Per raggiungere l'evento selezioniamo il box a destra di [figura 4](#) indicato dalla *freccia rossa* e nell'elenco a discesa scegliamo l'evento **Initialize** e inseriamo questo codice



```
Private Sub UserForm_Initialize ()  
    Anagrafica.Height = 100  
    Anagrafica.Width = 100  
End Sub
```

Avrete notato che per agire sulla Userform viene inserito il nome della stessa (Anagrafica) seguito dalla parola chiave Height (altezza) separate da un punto e con l'operatore uguale abbiamo fissato il valore a 100, la stessa operazione viene ripetuta per la larghezza con la parola chiave Width (larghezza). Se eseguiamo questa macro la Userform prenderà le dimensioni impostate, cioè sarà larga 100 pixel e alta di 100 pixel. Possiamo sfruttare un altro evento per modificare le dimensioni ogni volta che l'utente clicca sulla form aumentandole di 50 pixel ad ogni clic, inserendo questo codice nell'evento *Userform_Click*

```
Private Sub UserForm_Click ()  
    Anagrafica.Height = Anagrafica.Height + 50  
    Anagrafica.Width = anagrafica.Width + 50  
End Sub
```

Sempre con lo stesso metodo che abbiamo appena visto, abbiamo inserito il nome della form seguita dal punto e dalle istruzioni Height e Width e con le assegnazioni che abbiamo visto per le variabili è stato assegnato un valore di 50 pixel preceduto dall'operatore più (+). Un altro evento che dobbiamo prendere in considerazione è l'evento *Activate* che viene scatenato ogni volta che la form diventa attiva (cioè passa in primo piano). Si utilizza questo evento per aggiornare il contenuto dei controlli, in modo da riflettere i cambiamenti che possono essere avvenuti mentre la finestra non era attiva. Possiamo usare questo evento per preparare i vari controlli a ricevere i dati dell'utente, per esempio se siamo in presenza di un TextBox (casella di testo) o vari OptionButton (pulsante di selezione) possiamo fare in modo che quando viene attivata la form il **focus** (corrisponde ad avere il campo selezionato e pronto a ricevere il testo) sia già sul Textbox oppure di avere un Optionbox già selezionato. Vediamolo con un esempio

```
Private Sub UserForm_Activate()  
    TextBox1.SetFocus  
    OptionButton1.value=True  
End Sub
```

Il metodo più semplice per controllare un oggetto Userform è utilizzando i *Metodi* e le *Proprietà* predefinite della classe Userform e scrivere le procedure evento per la gestione della finestra e dei controlli in essa contenuti, i metodi più comuni da utilizzare sono: *Copy* – *Cut* – *Hide* – *Paste* – *PrintForm* – *Repaint* – *Show*

Hide : Nasconde la finestra di dialogo (la Userform) senza liberare la memoria associata all'oggetto, in questo modo vengono mantenuti i valori nei vari controlli contenuti in essa

Show : Rende visibile la finestra sullo schermo, se la finestra non è caricata in memoria, viene effettuato il caricamento.

Inoltre VBA fornisce due comandi che sono molto utili quando usiamo l'oggetto Userform, i comandi sono *Load* e *Unload*, questi comandi possono essere utilizzati per caricare l'oggetto in memoria e per liberare la memoria se occupata dall'oggetto. La sintassi per questi comandi è la seguente:

Load Oggetto
Unload Oggetto

In questo enunciato *Oggetto* rappresenta un riferimento valido ad un oggetto di tipo Userform, con il comando *Load* carichiamo l'oggetto in memoria, ma non lo rende visibile sullo schermo, e con il comando *Unload* lo scarichiamo dalla memoria.



Abbiamo appena detto che con Load carichiamo l'oggetto in memoria, ma non lo portiamo a video, inneschiamo solo l'evento *Initialize* della Userform ma per poter vedere l'oggetto sullo schermo dobbiamo usare il comando **Show**, l'enunciato è il seguente

Anagrafica.Show

Pertanto se vogliamo far comparire una finestra di dialogo a video dovremmo lanciare il comando *Show*, ma come facciamo? Una volta creata la Form abbiamo a disposizione solo metodi ed eventi, ma tutti riferiti all'oggetto Userform, mentre a noi serve un altro procedimento che veicoli la nostra finestra di dialogo. Se diamo uno sguardo alla finestra dei progetti vediamo che la Form è presente, ma abbiamo appena detto che deve essere veicolata per poter renderla visibile. Un sistema per ottenere questo è di seguire questo percorso **Inserisci - Modulo** e nella finestra di progetto ora ci comparirà anche il modulo

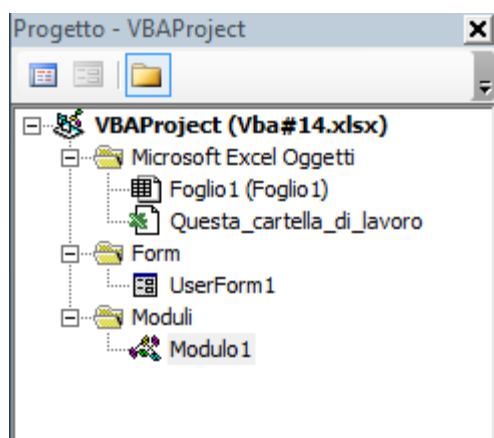


Fig. 5

A questo punto clicchiamo sulla voce "Modulo1" e nella finestra del codice digitiamo il seguente codice

```
Sub Mostra()  
Anagrafica.Show  
End Sub
```

Associamo ora la nostra macro ad un pulsante sul foglio di lavoro e premendo sul pulsante comparirà a video la nostra Userform. Ora abbiamo creato la nostra finestra di dialogo e siamo riusciti a portarla a video, fatte le nostre operazioni sulla form la possiamo "chiudere" usando il comando Unload. Il comando Unload va messo all'interno della Userform, associato ad un pulsante di uscita dalla stessa, la sintassi è la seguente:

Unload Me

Vediamo ora la finestra della *casella degli strumenti*, noterete che l'etichetta della finestra riporta il nome di *Controlli* e sono rappresentati dalle varie icone presenti nel box, vediamo cosa rappresentano e come si usano, per farvi comprendere meglio quali siano i controlli li ho raggruppati in questa immagine con il relativo nome e una breve descrizione



Icona	Comando	Italiano	Descrizione
	Select	Seleziona	Consente di selezionare altri controlli
	Label	Etichetta	Svolge la funzione di etichetta descrittiva non modificabile dall'utente
	TextBox	Casella di Testo	Permette di inserire del testo liberamente
	ComboBox	Casella Combinata	Permette di inserire un valore da una lista a scomparsa
	ListBox	Casella di Riepilogo	Mostra molti elementi in una lista con una barra di scorrimento verticale
	CheckBox	Casella di Controllo	Si utilizza per le opzioni che possono avere un valore tipo Vero/Falso
	OptionButton	Pulsante di Opzione	Consente all'utente di effettuare una scelta in una serie di pulsanti
	ToggleButton	Pulsante Interruttore	Mostra lo stato Vero/Falso e si presenta come un interruttore On/Off

Fig. 6

Icona	Comando	Italiano	Descrizione
	Frame	Cornice	Raggruppa una serie di controlli con un'etichetta descrittiva
	CommanButton	Pulsante di Comando	E' un pulsante standard per eseguire azioni
	TabStrip	Schede	tabulazione che può ricevere altri controlli
	Multipage	Pagine	Consiste in un'area di più pagine che può contenere altri controlli
	ScrollBar	Barra di Scorrimento	Permette di scorrere un insieme di dati
	SpinButton	Casella di Selezione	Particolare casella di testo che permette l'inserimento facilitato dei dati
	Image	Immagine	Permette di inserire un'immagine all'interno di una finestra
	RafEdit	Casella Speciale	E' una casella di testo speciale che consente di selezionare intervalli di celle

Fig. 7