



Il controllo Image e RafEdit

Un controllo **Image** consente di visualizzare delle immagini nella propria applicazione. Per assegnare una foto a questo controllo in fase di progettazione, si utilizza la proprietà Image, inoltre è possibile assegnare un'immagine al controllo Image in fase di esecuzione utilizzando la funzione [LoadPicture](#). Un aspetto importante è quello di inserire un'immagine nel controllo Image o adattare le dimensioni del controllo all'immagine, che determina come sarà visualizzata. Questo può essere fatto usando la proprietà [PictureSizeMode](#) come descritto di seguito.

La Proprietà **PictureSizeMode** specifica come viene visualizzata l'immagine, vale a dire, se il controllo Image non ha le stesse dimensioni dell'immagine è possibile modificare la modalità di visualizzazione con le proprietà disponibili che sono:

- *fmPictureSizeModeClip* (valore 0) - se il controllo immagine non ha le stesse dimensioni dell'immagine ritaglia le parti dell'immagine che non rientrano nello spazio del controllo immagine.
- *fmPictureSizeModeStretch* (valore 1) - ridimensiona l'immagine per adattarla al controllo Image. Se le proporzioni dell'immagine e del controllo sono diverse l'immagine viene distorta
- *fmPictureSizeModeZoom* (valore 3) - Ridimensiona l'immagine senza distorcerla e l'immagine viene ingrandita o rimpicciolita fino a che non raggiunge i margini orizzontali o verticali del controllo. Se il limite orizzontale viene raggiunto per primo, il rimanente spazio verticale apparirà vuoto e se il limite verticale viene raggiunto per primo, il rimanente spazio orizzontale apparirà vuoto.

La Proprietà *PictureSizeMode* può essere impostata nella finestra Proprietà oppure con codice VBA.

La Proprietà **PictureAlignment** specifica la posizione relativa o l'allineamento del quadro all'interno del controllo Image. Sono disponibili 5 impostazioni:

- *fmPictureAlignmentTopLeft* (valore 0)
- *fmPictureAlignmentTopRight* (valore 1)
- *fmPictureAlignmentCenter* (valore 2)
- *fmPictureAlignmentBottomLeft* (valore 3)
- *fmPictureAlignmentBottomRight* (valore 4).

Le impostazioni sono auto-esplicative, per esempio *fmPictureAlignmentBottomLeft* significa che l'immagine si allinea nell'angolo in basso a sinistra del controllo Image.



*Se l'immagine è impostata su *fmSizeModeStretch* nella proprietà *PictureSizeMode*, l'immagine si estende per adattarsi al suo controllo, quindi *PictureAlignment* non dispone di alcun effetto.*

Utilizzando la Funzione **LoadPicture** si assegna un'immagine al controllo Image in fase di esecuzione: si usa la seguente Sintassi: *ImageControl.Picture = LoadPicture (percorso)*



Esempio: Selezionare la foto da visualizzare nel controllo Image dal ListBox1

```
Private Sub UserForm_Initialize ()  
    ListBox1.AddItem "Immagine1"  
    ListBox1.AddItem "Immagine2"  
    ListBox1.AddItem "Immagine3"  
    ListBox1.AddItem "Immagine4"  
End Sub  
  
Private Sub ListBox1_Click ()  
    If ListBox1.Value = "immagine 1" Then  
        Image1.Picture = LoadPicture ("C:\Test\Immagini\immagine1.jpg")  
    ElseIf ListBox1.Value = "immagine2" Then  
        Image1.Picture = LoadPicture ("C:\Test\Immagini\immagine2.jpg")  
    ElseIf ListBox1.Value = "immagine3" Then  
        Image1.Picture = LoadPicture ("C:\Test\Immagini\immagine3.jpg")  
    ElseIf ListBox1.Value = "immagine4" Then  
        Image1.Picture = LoadPicture ("C:\Test\Immagini\immagine4.jpg")  
    End If  
End Sub
```

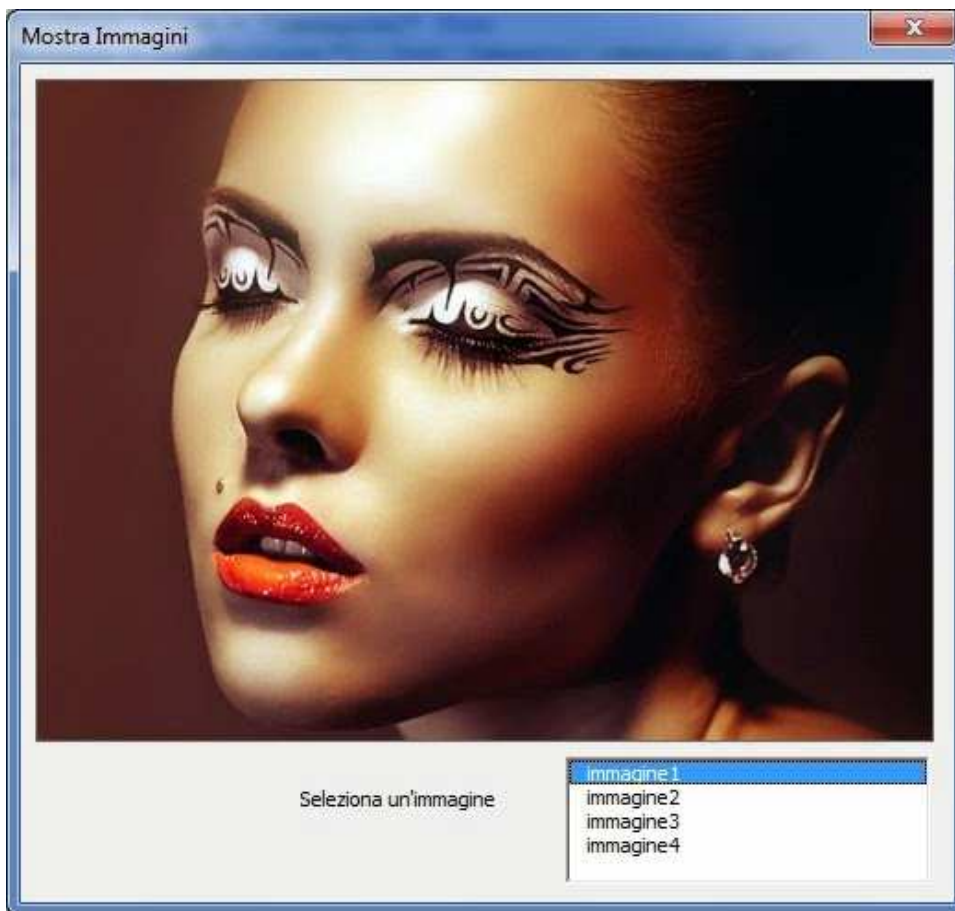


Fig. 1



Esempio: Selezionare il percorso completo di un'immagine da visualizzare nel controllo Image dal ListBox1

```
Private Sub UserForm_Initialize ()  
Dim img1 As String  
Dim img2 As String  
Dim img3 As String  
Dim img4 As String  
  
img1 = "C:\Test\Immagini\immagine1.jpg"  
img2 = "C:\Test\Immagini\immagine2.jpg"  
img3 = "C:\Test\Immagini\immagine3.jpg"  
img4 = "C:\Test\Immagini\immagine4.jpg"  
  
ListBox1.AddItem img1  
ListBox1.AddItem img2  
ListBox1.AddItem img3  
ListBox1.AddItem img4  
End Sub  
  
Private Sub ListBox1_Click ()  
Image1.Picture = LoadPicture (ListBox1.List (ListBox1.ListIndex))  
End Sub
```

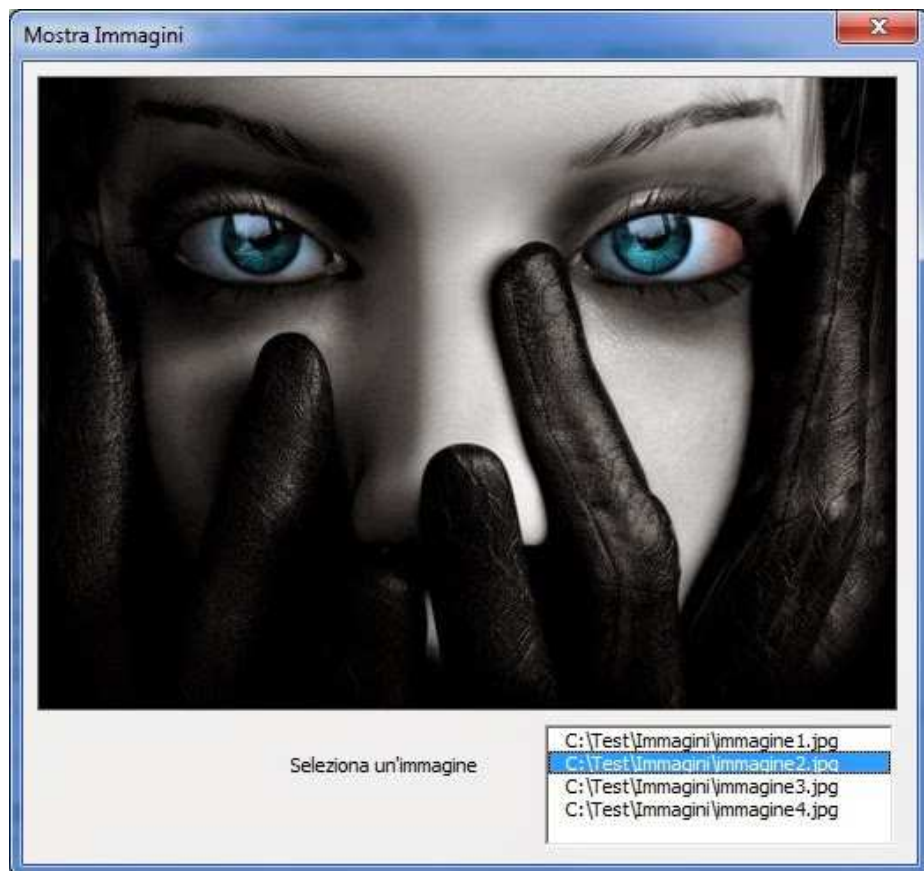


Fig. 2



Esempio: Visualizzare un'immagine nel pulsante di comando, all'attivazione della UserForm:

```
Private Sub UserForm_Initialize ()  
  CommandButton1.Picture = LoadPicture ("C:\Test\Immagini\img1.jpg")  
End Sub
```



Fig. 3

Esempio: Visualizzare un immagine nel controllo Image cliccando sul pulsante di comando:

```
Private Sub CommandButton1_Click ()  
  With UserForm2  
    Image1.Picture = LoadPicture ("C:\Test\Immagini\immagine4.jpg")  
  End With  
End Sub
```



Fig. 4



■ Metodo GetOpenFilename

Questo metodo consente all'utente di immettere un nome di file nella finestra di dialogo standard *Apri File*. Il metodo non apre qualsiasi file, viene visualizzata solo la casella per accettare il nome del file, e restituisce il nome inserito dall'utente o il nome del file selezionato, che potrebbe anche essere un percorso (cioè cartella e nome del file). La sintassi è la seguente:

Espressione GetOpenFilename ([FileFilter], [FilterIndex], [Titolo], [ButtonText], [Selezione multipla])

FileFilter è una stringa che specifica i criteri per filtrare i file. Si compone di coppie di stringhe e ogni parte comprende il carattere jolly ed è separato da virgole. Esempio: *FileFilter*: = "File di testo (txt.), * txt, TIFF (* tif.), * tif, JPEG Files (* jpg.), * jpg"

L'esempio seguente di *FileFilter* specifica 3 criteri di filtri di file (file di testo txt, i file TIFF e file JPEG) e consente all'utente di inserire o selezionare i file con estensione txt, Jpg e tif. Per includere più espressioni in un unico criterio di filtro di file, utilizzare punti e virgola per separare ogni espressione.

Esempio: *FileFilter*: = "JPEG File Interchange Format (* jpg, * jpeg, * jfif; * GPE), * jpg, * jpeg, * jfif; * GPE"

E' possibile utilizzare *FileFilter* specificando 1 criterio solo di filtro per i file (JPEG) e comprende quattro espressioni che consentono la selezione di file con estensione jpg, jpeg, jfif e GPE. Tralasciando questo argomento sarà di default il filtro "Tutti i file (*. *), *. *".

FilterIndex è il criterio di default del filtro dei file specificati dal numero indice che va da 1 al numero di criteri specifici di filters. Omettendo di menzionare *FilterIndex*, o specificando un numero di indice oltre la sua dotazione, verrà impostato il primo criterio del filtro file.

Titolo specifica il titolo della finestra di dialogo che viene visualizzata utilizzando il metodo *GetOpenFilename*. Il titolo predefinito è "Open" se non specificato.

MultiSelect se impostato a True consente la selezione multipla dei file e se impostato a False, che è anche il valore di default, permette la selezione di un singolo file.

Esempio: Usare *LoadPicture* con il metodo *GetOpenFilename* per caricare foto in un controllo Image.

```
Private Sub UserForm_Initialize()  
Me.Image1.BackColor = RGB(152, 255, 152)  
Me.Label1.Caption = "Clicca sulla casella verde per selezionare e caricare un'Immagine"  
Me.CommandButton1.Caption = "Cliccare per Uscire"  
End Sub  
  
Private Sub Image1_Click()  
Dim fileName As String  
fileName = Application.GetOpenFilename(filefilter:="Tiff Files(*.tif;*.tiff),*.tif;*.tiff,JPEG  
Files (*.jpg;*.jpeg;*.jfif;*.jpe),*.jpg;*.jpeg;*.jfif;*.jpe,Bitmap Files(*.bmp),*.bmp",  
FilterIndex:=2, Title:="Seleziona un file", MultiSelect:=False)  
If fileName = "False" Then  
MsgBox "File non selezionato!"  
Else  
Me.Image1.Picture = LoadPicture(fileName)  
Me.Repaint  
Me.Label1.Caption = "Immagine Caricata"  
End If  
End Sub
```

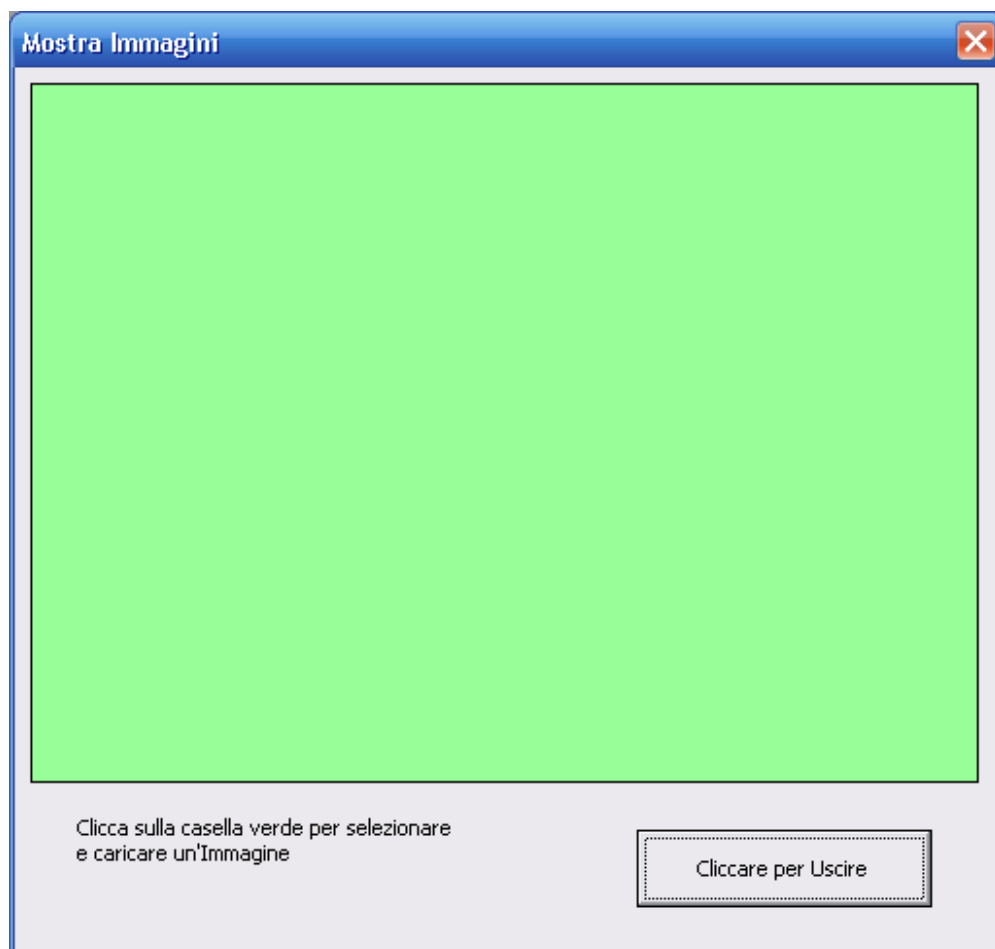


Fig. 5



Fig. 6

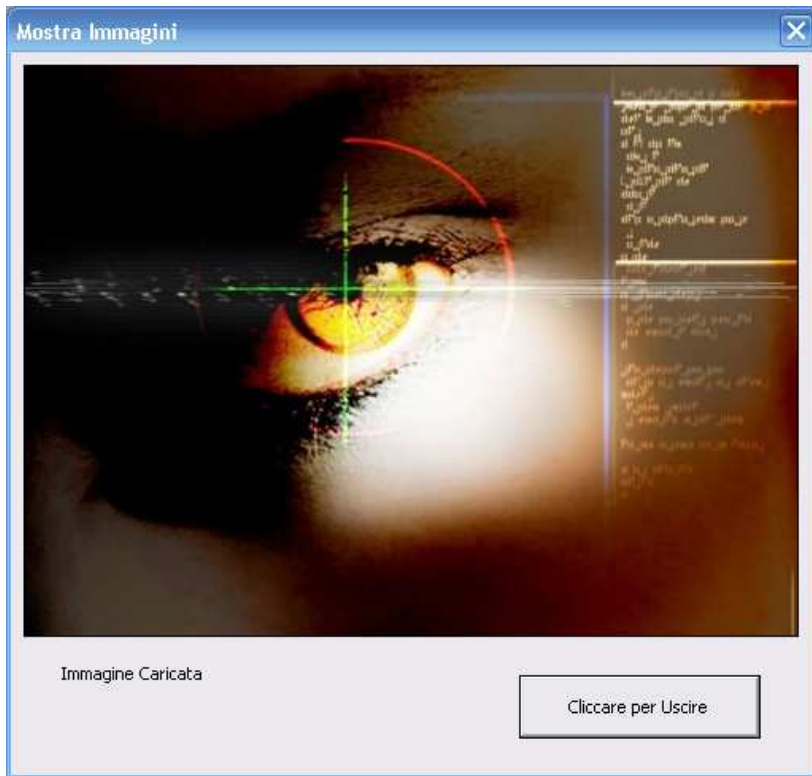


Fig. 7

Esempio: Usare LoadPicture con il metodo GetOpenFilename per caricare foto in un controllo Image

```
Private Sub UserForm_Initialize()
Me.Image1.BackColor = RGB(152, 255, 152)
Me.Label1.Caption = "Clicca sulla casella verde per selezionare e caricare un'Immagine"
Me.CommandButton1.Caption = "Cliccare per Uscire"
End Sub

Private Sub Image1_Click()
Dim fltr As String, ttl As String, fileName As String
Dim fltrIndx As Integer
Dim mltiSlct As Boolean
fltr = "Tiff Files(*.tif;*.tiff),*.tif;*.tiff,JPEG Files"
(*.jpg;*.jpeg;*.jfif;*.jpe),*.jpg;*.jpeg;*.jfif;*.jpe,Bitmap Files(*.bmp),*.bmp"
fltrIndx = 2
ttl = "Seleziona un file"
mltiSlct = False
ChDrive "C"
ChDir "C:\Test\Immagini"
fileName = Application.GetOpenFilename(fltr, fltrIndx, ttl, , mltiSlct)
If fileName <> "False" Then
Me.Image1.Picture = LoadPicture(fileName)
Me.Repaint
Me.Label1.Caption = "Immagine Caricata"
End If
End Sub

Private Sub CommandButton1_Click()
Unload Me
End SubPrivate Sub CommandButton1_Click()
Unload Me
End Sub
```



Il risultato ottenuto è lo stesso visibile nelle Figure 5-6-7.

■ Il Controllo RefEdit

Il controllo *RefEdit* è disponibile solo in una UserForm e consente all'utente di selezionare una cella o un intervallo di celle del foglio di lavoro, permette anche di digitare direttamente nella casella di testo i riferimenti di cella. Nel campo del controllo RefEdit è visibile l'indirizzo della cella o l'intervallo di celle, che un utente seleziona o digita nel campo stesso. Per ottenere l'indirizzo di una cella o l'intervallo di celle memorizzate in un controllo RefEdit si utilizza la proprietà Value.

Esempio: Selezionare una cella nel foglio ed eseguire il codice premendo un pulsante

```
Private Sub CommandButton1_Click ()  
Dim rngAddress As String  
rngAddress = RefEdit1.Value  
Range (rngAddress). Value = "Ciao"  
Range (rngAddress). Interior.Color = RGB (255, 0, 0)  
If MsgBox ("Azione eseguita. Vuoi Uscire?", VbQuestion + vbYesNo) = vbYes Then  
Unload Me  
End If  
End Sub
```



Fig. 8

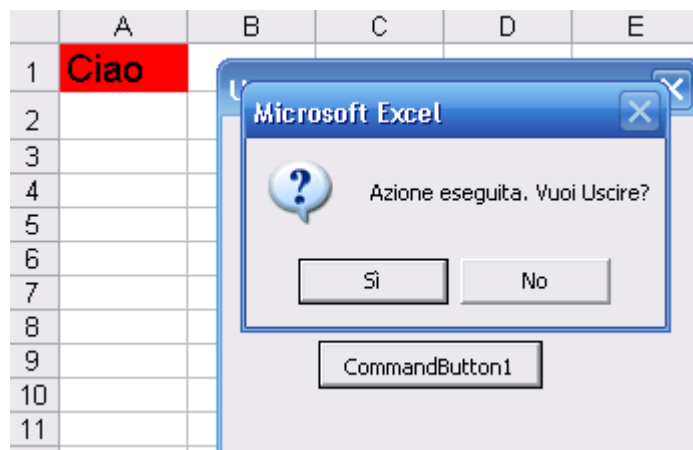


Fig. 9