



Visualizza la finestra di dialogo per la selezione di una cartella




http://www.vbsimple.net/howto/ht_009.htm

Difficoltà:  3 / 5

Insieme alla versione Professional di Visual Basic è fornito il controllo Microsoft Common Dialog, utile per la selezione di un file da aprire o salvare. Ma non esiste un modo per selezionare una cartella tramite queste finestre.

Esiste a questo scopo una finestra di dialogo all'interno della DLL *Shell32*. Il suo nome è **BrowseForFolder**. In questo HowTo vedremo come richiamare questa funzione mediante l'[API](#).

Per il nostro scopo creeremo una funzione che ritorna il percorso, ma prima è necessario scrivere alcune dichiarazioni API all'interno del nostro form oppure in un modulo standard .




```

1. Option Explicit
2.
3. Private Type BROWSEINFO
4.     hOwner As Long
5.     pidlRoot As Long
6.     pszDisplayName As String
7.     lpszTitle As String
8.     ulFlags As Long
9.     lpfn As Long
10.    lParam As Long
11.    iImage As Long
12. End Type
13.
14. Private Const BIF_RETURNONLYFSDIRS = &H1
15. Private Declare Function SHBrowseForFolder Lib "shell32.dll" Alias
    "SHBrowseForFolderA" (lpBrowseInfo As BROWSEINFO) As Long
16. Private Declare Function SHGetPathFromIDList Lib "shell32.dll" Alias
    "SHGetPathFromIDListA" (ByVal pidl As Long, ByVal pszPath As String) As Long
17. Private Declare Sub CoTaskMemFree Lib "ole32.dll" (ByVal pv As Long)

```

Le righe dal 3 al 12 definiscono un nuovo tipo di nome **BROWSEINFO**. Tale struttura è richiesta dalla funzione API.

La riga 14 presenta una [costante](#)  **BIF_RETURNONLYFSDIRS** che indica che vogliamo presentare soltanto le cartelle. Ulteriori costanti che è possibile utilizzare in questa funzione sono presenti nelle dichiarazioni API. Per rintracciarle bisogna cercare tutte le costanti il cui nome inizia per "**BIF_**" (BrowseInfo Flags).

Alla riga 15 viene dichiarata la prima funzione API: **SHBrowseForFolders**. Essa richiede che sia passata una variabile di tipo BROWSEINFO (definita prima), mostra la finestra di dialogo per la scelta della cartella e, nel momento in cui l'utente clicca il pulsante OK

[alloca](#) in uno spazio apposito il nome della cartella selezionata.

La seconda funzione API (riga 16) estrae il percorso della cartella selezionata dall'area di memoria allocata.

L'ultima funzione API (riga 17) [dealloca](#) l'area di memoria bloccata dalla funzione `SHBrowseForFolders`.

Terminate le dichiarazioni passiamo alla scrittura di una funzione che faccia tutto il lavoro mediante un semplice richiamo. La nostra funzione prende il nome di *SfoggiaCartelle*.

```
19. Public Function SfoggiaCartelle(ByVal Handle As Long) As String
```

La funzione richiede l'[handle](#) del form dentro il quale apparirà la finestra di scelta della cartella e restituisce una stringa in uscita.

```
20.     Dim BInfo As BROWSEINFO
```

Come già accennato prima, la funzione *SHBrowseForFolders* richiede che sia passata una variabile con la struttura `BROWSEINFO`. La nostra variabile si chiamerà **BInfo**.

```
21.     Dim CartellaScelta As Long
22.     Dim PercorsoAllocato As Long
23.     Dim PercorsoScelto As String
```

Dichiareremo anche una serie di altre variabili: **CartellaScelta** indica che è stato possibile estrarre il percorso scelto dall'area di memoria allocata. In caso di mancata estrazione - ovvero quando l'utente preme il tasto *Annulla* della finestra di dialogo - essa avrà il valore 0.

La variabile **PercorsoAllocato** conterrà l'indirizzo dell'area di memoria allocata dalla funzione. Tale valore sarà in seguito utilizzato per ricavare il percorso della cartella selezionata che verrà memorizzato nella variabile **PercorsoScelto**.

```
24.     BInfo.hOwner = Handle
25.     BInfo.lpszTitle = "Scegli una cartella:"
26.     BInfo.ulFlags = BIF_RETURNONLYFSDIRS
```

La struttura `BROWSEINFO` ha tre elementi fondamentali: *hOwner* che indica l'handle della finestra dentro la quale apparirà la nostra finestra di dialogo. La nostra funzione **SfoggiaCartelle** ci provvede l'handle da assegnare a tale elemento (riga 24). Altro elemento fondamentale è *lpszTitle* e indica il testo che apparirà nella parte superiore della finestra di dialogo. L'ultimo elemento è anche il più importante: esso indica il tipo di azione da eseguire con tale finestra: noi assegneremo il valore della costante **BIF_RETURNONLYFSDIRS** dichiarata prima.

```
27.     PercorsoAllocato = SHBrowseForFolder(BInfo)
```

Eseguiamo la chiamata alla funzione *SHBrowseForFolders* che mostrerà la finestra di scelta della cartella. All'uscita della finestra, la variabile **PercorsoAllocato** conterrà un [puntatore](#) all'area di memoria allocata dall'ultima funzione.

```
28.     PercorsoScelto = Space$(512)
```

Come per tutte le funzioni API che utilizzano stringhe, è necessario preparare prima la variabile mediante l'istruzione *Space\$*.

```
29.      CartellaScelta = SHGetPathFromIDList(PercorsoAllocato, PercorsoScelto)
```

La funzione **SHGetPathFromIDList** estrae il percorso allocato dall'area di memoria indicata da PercorsoAllocato e lo memorizza nella variabile PercorsoScelto.

Se l'utente ha annullato la scelta, il tentativo di estrazione del percorso ritornerà il valore 0.

```
30.      If CartellaScelta Then
```

Quindi segue un controllo che si assicura che l'utente abbia scelto qualche cartella. In caso negativo il programma salta alla riga 34.

```
31.      SfoggiaCartelle = Left$(PercorsoScelto, InStr(PercorsoScelto, Chr$(0)) - 1)
```

Tutte le funzioni API restituiscono le variabili stringa con un terminatore (codice ASCII 0) alla fine. Questa riga provvede a eliminare il terminatore finale da **PercorsoScelto** e memorizza il rimanente nella variabile **SfoggiaCartelle** che sarà il valore ritornato all'uscita della funzione.

```
32.      CoTaskMemFree PercorsoAllocato
```

Fatto questo è necessario deallocare la memoria allocata.

La funzione **CoTaskMemFree** provvede a questo e richiede un puntatore.

```
33.      Else
```

```
34.      SfoggiaCartelle = ""
```

L'utente ha annullato la selezione. Non sarà ritornata alcuna cartella all'uscita.

```
35.      End If
```

```
36. End Function
```

```
37.
```

Terminata la nostra funzione possiamo passare alla semplice interfaccia.

Inseriamo sopra il nostro form due controlli: un pulsante di comando di nome **Sfoggia** ed una casella di testo di nome **Percorso**.



Nel momento in cui l'utente clicca il pulsante Sfoggia viene richiamato l'evento click:

```
38. Private Sub Sfoggia_Click()
```

```
39.      Percorso.Text = SfoggiaCartelle(Me.hWnd)
```

```
40. End Sub
```

Questa semplice routine scrive il percorso della cartella selezionata all'interno della casella di testo di nome **Percorso**.

È terminata qui. Basta cliccare il pulsante Sfoggia per vedere apparire la finestra di dialogo "Sfoggia per Cartelle".



[Torna all'indice degli HowTo](#)
