Visualizzare la finestra delle proprietà di un file

http://www.vbsimple.net/howto/ht_024.htm

Difficoltà: 2 / 5

All'interno di Windows tutti i files e le cartelle hanno una finestra delle proprietà dove vengono mostrati i dati dell'elemento scelto, quali la dimensione, il percorso all'interno del file system, il tipo di oggetto e, per alcuni tipi di files, finestre aggiuntive.

Normalmente per accedere alla finestra delle proprietà di un file basterà cliccare con il tasto destro del mouse sul file e scegliere la voce ***Proprietà***.
Per effettuare la medesima cosa attraverso Visual Basic sarà necessario utilizzare la funzione API *ShellExecuteEX* che permette di eseguire determinate operazioni della Shell di Windows.

Vedremo in questo semplice HowTo come richiamare la finestra delle proprietà di un file attraverso la suddetta funzione API.
In questo progetto sarà necessario un form ed un modulo standard in cui risiederanno le chiamate all'API.

Cominceremo proprio dal modulo standard:

```
 1. Option Explicit
 2.
 3. Private Type SHELLEXECUTEINFO
 4.     cbSize       As Long
 5.     fMask        As Long
 6.     hwnd         As Long
 7.     lpVerb       As String
 8.     lpFile       As String
 9.     lpParameters As String
10.     lpDirectory  As String
11.     nShow        As Long
12.     hInstApp     As Long
13.     lpIDList     As Long
14.     lpClass      As String
15.     hkeyClass    As Long
16.     dwHotKey     As Long
17.     hIcon        As Long
18.     hProcess     As Long
19. End Type
20.
21. Private Const SEE_MASK_INVOKEIDLIST = &HC
22. Private Const SEE_MASK_NOCLOSEPROCESS = &H40
23. Private Const SEE_MASK_FLAG_NO_UI = &H400
24.
25. Private Declare Function ShellExecuteEx Lib "SHELL32.DLL" (SEI As SHELLEXECUTEINFO)
    As Long
26.
```

Alla riga 3 abbiamo definito un nuovo tipo di dati di nome **SHELLEXECUTEINFO**.
Tale struttura è richiesta dalla funzione API.

Alle righe 21-23 abbiamo dichiarato tre costanti che utilizzeremo come parametri nella

struttura SHELLEXECUTEINFO.

Infine, alla riga 25, abbiamo la dichiarazione della funzione ***ShellExecuteEx***. Essa differisce dalla funzione *ShellExecute*, già vista in un altro HowTo ed in una richiesta dei lettori, per il fatto che permette di specificare una serie di parametri aggiuntivi che l'altra funzione non permette.

```
27. Public Function ShowPropertiesDlg(ByVal FileName As String, ByVal OwnerhWnd As
    Long) As Long
28.     Dim SEI As SHELLEXECUTEINFO
29.     With SEI
30.         .cbSize = Len(SEI)
31.         .fMask = SEE_MASK_NOCLOSEPROCESS Or SEE_MASK_INVOKEIDLIST Or
    SEE_MASK_FLAG_NO_UI
32.         .hWnd = OwnerhWnd
33.         .lpVerb = "properties"
34.         .lpFile = FileName
35.         .lpParameters = vbNullChar
36.         .lpDirectory = vbNullChar
37.         .nShow = 0
38.         .hInstApp = 0
39.         .lpIDList = 0
40.     End With
```

La nostra funzione si chiamerà **ShowPropertiesDlg** e richiede due parametri: il nome del file per il quale mostrare la finestra delle proprietà e l'handle della finestra chiamante.

Alla riga 28 abbiamo dichiarato la variabile **SEI** di tipo *SHELLEXECUTEINFO*. Essa accoglierà tutti i parametri da passare alla funzione ShellExecuteEx.

Così nelle righe 29-40 definiamo tali parametri: ***cbSize*** indica la dimensione della struttura, calcolata tramite l'istruzione Len; ***fMask*** definisce le modalità di visualizzazione e richiamo della funzione ed inseriremo i tre valori dichiarati prima come costanti; ***hWnd*** riceverà l'handle passato alla funzione; ***lpVerb*** richiede il nome dell'operazione da eseguire (*properties*); ***lpFile*** indica il nome del file per il quale eseguire l'operazione *lpVerb*. Tutti gli altri dati della struttura non saranno utilizzati in questa funzione, pertanto li lasceremo in bianco, cioè impostati a 0.

```
41.     If (ShellExecuteEx(SEI) = 0) Then
42.         ShowPropertiesDlg = 0
43.     Else
44.         ShowPropertiesDlg = SEI.hInstApp
45.     End If
46. End Function
```

Finalmente alla riga 41 abbiamo la chiamata alla funzione *ShellExecuteEx* passandole come parametro la struttura **SEI**. Se la funzione ritorna valore 0, tale sarà anche il valore ritornato dalla nostra funzione *ShowPropertiesDlg*; se la funzione ritorna un valore diverso da 0, sarà necessario leggere il contenuto del campo ***hInstApp*** della struttura SEI.

Terminata la scrittura del modulo standard possiamo passare alla progettazione del form che richiamerà la funzione appena vista.

Figura 1

Il form si compone di tre semplici controlli: una *Label* di nome **Label1**, una *TextBox* di nome **txtFileName** ed un *CommandButton* di nome **btnProperties**.

L'utilizzo dell'interfaccia è ovvio: basterà inserire il nome di un file nella casella di testo e premere il pulsante ***"Proprietà del file"*** per vedere apparire la finestra delle proprietà corrispondente. Prima di fare questo, è necessario scrivere il codice che controlla la pressione del pulsante **btnProperties**.

```
1. Option Explicit
2.
3. Private Sub btnProperties_Click()
4.     Dim ret As Long
5.     ret = ShowPropertiesDlg(txtFileName.Text, Me.hWnd)
6.     If (ret <= 32) Then MsgBox "Errore!", vbCritical, "Errore"
7. End Sub
```

La routine che regola l'evento Click del pulsante si compone di tre semplici righe.
Alla riga 4 dichiariamo la variabile **ret** che riceverà il valore restituito dalla chiamata della funzione *ShowPropertiesDlg* con il testo della casella **txtFileName** e l'handle della finestra (riga 5).

Se il valore restituito dalla funzione è minore o uguale di 32, siamo in presenza di qualche errore, ad esempio il nome del file specificato non esiste. Se tale valore è minore o uguale di 32, verrà mostrata un messaggio di errore (riga 6).

Proviamo il nostro progetto.

Un click sopra il pulsante ***"Proprietà del file"*** mostrerà la finestra delle proprietà corrispondente.

Nel caso avessimo inserito un nome di file errato, sarebbe apparso invece il messaggio di errore.

La separazione della funzione ***ShowPropertiesDlg*** in un modulo standard rende molto semplice il suo utilizzo all'interno di altri progetti.

Roal Zanazzi
2 Febbraio 2001


Torna all'indice degli HowTo

