Elencare i files di una cartella secondo un criterio (seconda parte)

http://www.vbsimple.net/howto/ht_014_2.htm

[<< Continua dalla parte 1]

Seconda versione (con l'utilizzo dell'API)

Difficoltà: 4 / 5

Una soluzione alternativa consiste nello sfruttamento di alcune funzioni API, in particolare l'invio di un messaggio ben definito alla *ListBox* che dovrà contenere i nomi dei files.

Il messaggio in questione è ***LB_DIR*** applicato alla ListBox. Per cui è necessario definire alcune costanti e dichiarazioni API:

```
1. Private Const LB_DIR = &H18D
2. Private Const DDL_ARCHIVE = &H20
3. Private Const DDL_HIDDEN = &H2
4. Private Const DDL_READONLY = &H1
5. Private Const DDL_SYSTEM = &H4
6. Private Declare Function SendMessage Lib "USER32" Alias "SendMessageA" (ByVal hwnd
   As Long, ByVal wMsg As Long, ByVal wParam As Long, lParam As Any) As Long
7. Private Declare Function GetShortPathName Lib "KERNEL32" Alias
   "GetShortPathNameA" (ByVal lpszLongPath As String, ByVal lpszShortPath As String,
   ByVal cchBuffer As Long) As Long
```

La prima costante è proprio il messaggio ***LB_DIR*** che provvederà ad effettuare il riempimento della ListBox.
Seguono alcune costanti di tipo DDL_ che definiscono il tipo di file da elencare: *DDL_ARCHIVE* indica i files il cui attributo di archivio è impostato; *DDL_HIDDEN* indica i files nascosti; *DDL_READONLY* specifica i files di sola lettura; infine *DDL_SYSTEM* indica i files di sistema.

Nota che i valori DDL_ specificati corrispondono ai valori vbArchive, vbHidden, vbReadOnly e vbSystem dell'enumerazione *VbFileAttribute*. Ciò significa che avremmo potuto fare a meno di queste costanti ed utilizzare il tipo intrinseco di Visual Basic.

Alla riga 6 abbiamo dichiarato la funzione API *SendMessage*, utilizzata per inviare un messaggio ad un oggetto. La funzione richiede quattro parametri:

- **hWnd**: handle della finestra (oggetto) a cui inviare il messaggio;
- **wMsg**: messaggio da inviare alla finestra;

- **wParam**: valore numerico per specificare un parametro per il messaggio;
- **lParam**: valore generico per specificare un ulteriore parametro.

Alla riga 7 dichiariamo la funzione API *GetShortPathName* che converte un nome di file lungo nel formato DOS (8.3). Esiste un *HowTo* che spiega passo passo il funzionamento di questa funzione API; per tale ragione essa non sarà approfondita in questa sede.

Per utilizzare questa nuova funzione di elencazione files inseriamo un nuovo pulsante di comando di nome **ElencaAPI** sul nostro form.

Purtroppo però questa nuova soluzione non permette la scansione delle sottocartelle.

Quindi l'impostazione del valore della *CheckBox* non varia nulla. In ogni caso **non saranno scandite** le sottocartelle.

```
 9. Private Sub ElencaAPI_Click()
10.     Dim TempoInizio As Single
11.     Dim TempoFine As Single
12.     Dim Percorso As String
13.     ElencaAPI.MousePointer = vbHourglass
14.     TempoInizio = Timer
15.     ListaFiles.Clear
```

Fino a questo punto la funzione differisce dalla precedente solo per la variabile stringa **Percorso** dichiarata alla riga 12. Essa servirà per ottenere il percorso della cartella da analizzare.

```
16.     Percorso = String(Len(CercaText.Text) + 1, 0)
17.     Percorso = Left$(Percorso, GetShortPathName(ByVal CercaText.Text, Percorso, Len
    (CercaText.Text) + 1))
```

Queste due righe sono l'applicazione (anche se un po' più complessa) della funzione API *GetShortPathName* vista in un altro *HowTo*. Esse convertono il percorso specificato nella *TextBox* **CercaText** in formato DOS e memorizzano il risultato nella variabile **Percorso**.

```
18.     If Right(Percorso, 1) <> "\" Then Percorso = Percorso & "\"
19.     Percorso = Percorso & TipoText.Text
```

La riga 19 concatena il tipo di file ricercato al percorso, ma prima si assicura (riga 18) che la variabile **Percorso** termini con un \ e provvede ad aggiungerlo nel caso contrario.

```
20.     Call SendMessage(ListaFiles.hwnd, LB_DIR, DDL_ARCHIVE Or DDL_HIDDEN Or
    DDL_READONLY Or DDL_SYSTEM, ByVal Percorso)
```

In quest'unica riga risiede l'intera funzione di elencazione.
Viene richiesto di inviare il messaggio **LB_DIR** alla finestra il cui handle è ListaFiles.hwnd, ovvero la *ListBox* **ListaFiles**.
Vengono inviati due parametri: il primo è un numero composto dall'operazione di OR tra i valori DDL; in tal modo verranno ricercati tutti i files, ovvero quelli con l'attributo archivio, quelli nascosti, quelli di sola lettura e quelli di sistema.

Il secondo parametro è la variabile **Percorso**, passata, però per valore, come richiesto dalla maggior parte delle funzioni API. Essa, lo ricordiamo, è composta dal percorso della cartella da analizzare e dal tipo di file richiesto (vedi riga 19).

Questa semplice chiamata riempie la ListBox con i nomi dei files corrispondenti al criterio richiesto.

```
21.      TempoFine = Timer
22.      Tempo.Caption = "Tempo: " & TempoFine - TempoInizio
23.      Conteggio.Caption = "Conteggio: " & ListaFiles.ListCount
24.      ElencaAPI.MousePointer = vbNormal
25.  End Sub
```

La funzione si chiude con le stesse righe che completano la versione precedente. Viene scritta la durata dell'elencazione nella *Label* **Tempo** e viene scritto nella *Label* **Conteggio** il numero di elementi di cui si compone la ListBox.

Terminata la scrittura della funzione possiamo passare alla prova:

Figura 2

Questa funzione è leggermente più complessa della versione precedente, ma richiede un tempo di elaborazione nettamente inferiore a quello dell'altra versione.
Provare per credere! :)
Il grosso svantaggio è che non è possibile elencare i files presenti nelle sottocartelle tramite una sola chiamata.

Fibia FBI e Giuseppe Della Bianca
27 Dicembre 2000

Torna all'indice degli HowTo

