
[Home Page](#)
[Informazioni](#)
[Aiuto](#)

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

[http://www.vbsimple.net/howto/ht\\_014.htm](http://www.vbsimple.net/howto/ht_014.htm)

- [Prima versione \(senza l'utilizzo dell'API\)](#)
- [Seconda versione \(con l'utilizzo dell'API\)](#)

Questa volta vedremo come elencare i files di una cartella e delle sue sottocartelle, rispondenti ad un determinato criterio quali i [caratteri jolly \(wildcards\)](#).

### Prima versione (senza l'utilizzo dell'API)

Difficoltà: 3 / 5

Il metodo risolutivo di questa funzione si basa su una prima raccolta delle sottocartelle contenute nel percorso selezionato, ed in seguito a questo viene effettuata la raccolta dei files all'interno d'esse. La funzione così non è ottimizzata, ma è un buon compromesso tra prestazioni e semplicità.

Prima di vedere la funzione di elencazione dei files, progettiamo una semplice interfaccia grafica per mettere in atto la ricerca.

Inseriamo sopra il nostro form alcuni controlli:

1. Una **Label A** di nome **Tempo**, che indicherà il tempo di processamento della funzione. Non è obbligatoria, ma può essere utile per monitorare l'effettiva ottimizzazione della funzione.
2. Una **Label A** di nome **Conteggio**, che conterrà il numero di files trovati.
3. Una **ListBox** di nome **ListaFiles**, che conterrà l'elenco dei files trovati.
4. Una **Label A** di nome **CercaLabel**, contenente il testo "**Cerca nella cartella**". Essa servirà soltanto a spiegare la funzione della casella di testo sottostante.
5. Una **TextBox** di nome **CercaText** che l'utente utilizzerà per specificare il percorso da cui elencare i files.
6. Una **Label A** di nome **TipoLabel** contenente il testo "**Tipo File**" che spiegherà la funzione della casella sottostante.
7. Una **TextBox** di nome **TipoText** che conterrà il criterio utilizzato per l'elencazione dei nomi dei files.



8. Una *CheckBox*  di nome **SottoCartelle** utile a includere/escludere le sottocartelle nell'elencazione dei files.
9. Un *CommandButton*  di nome **Elenca** contenente il testo "**Elenca Files**". Naturalmente il click sopra d'esso effettuerà la scansione delle cartelle e l'elencazione dei files.

Prima di vedere la funzione di scansione dichiariamo due semplici [costanti](#)  entrambe di tipo Integer, che verranno utilizzate in seguito.

1. Option Explicit
2. Private Const TrovaTutto As Integer = vbHidden + vbSystem + vbDirectory
3. Private Const TrovaFiles As Integer = vbHidden + vbSystem

La prima costante è **TrovaTutto** ed il suo valore è dato dalla somma dei valori *vbHidden*, *vbSystem* e *vbDirectory*. Il suo utilizzo avviene con la chiamata dell'istruzione *Dir*. Infatti tale istruzione richiede, opzionalmente, il tipo di dato da ricercare. Passando tale valore all'istruzione *Dir* vengono estratti sia files che directory (cartelle). Purtroppo tale impostazione è d'obbligo in quanto la semplice specificazione di *vbDirectory* quale tipo di elemento da ricercare, riporterebbe **soltanto** le directory non nascoste e non di sistema, mentre salterebbe totalmente le cartelle nascoste e di sistema.

La seconda costante, molto simile alla prima, è **TrovaFiles**. Il suo utilizzo è analogo a quello della precedente costante, ma la somma dei valori di cui si compone la costante, fa sì che l'istruzione *Dir*, con tali valori, estragga **soltanto** i files ed eviti completamente le cartelle.

La prima costante verrà utilizzata per estrarre le cartelle dal percorso di partenza, mentre la seconda verrà sfruttata per elencare i files all'interno d'esse.

Vediamo quindi la nostra funzione di scansione: il suo nome sarà **ElencaFiles** e la sua dichiarazione è la seguente:

```
Private Sub ElencaFiles(PercSorg As String, TipoFile As String, InSottoDir As Boolean)
```

La funzione in realtà è una Sub, per cui non ritorna alcun valore in uscita; vedi a tal proposito le [informazioni aggiuntive sulla funzioni](#).

Richiede tre parametri: **PercSorg**, una stringa indicante il percorso della cartella da cui iniziare la scansione, **TipoFile**, stringa per specificare un filtro, un criterio, in base a [caratteri jolly](#) e **InSottoDir** di tipo booleano, che indica se la scansione debba ricercare i files contenuti anche nelle sottocartelle della cartella specificata con **PercSorg**.

5. Private Sub ElencaFiles(PercSorg As String, TipoFile As String, InSottoDir As Boolean)
6. Dim ListaDir As New Collection
7. Dim PercAtt As String
8. Dim NomeFile As String
9. Dim FileAtt As String
10. Dim DirNum As Integer

All'interno della funzione abbiamo dichiarato - ed istanziato mediante la parola *New* - una Collection di nome **ListaDir**. Essa verrà utilizzata per contenere i nomi di tutte le cartelle

che dovranno essere analizzate.

Abbiamo dichiarato anche una serie di variabili: **PercAtt**, stringa che conterrà il percorso della cartella al momento analizzata, estratta dalla Collection **ListaDir**; **NomeFile** invece servirà come variabile di appoggio per l'elencazione dei files e delle cartelle tramite il comando **Dir**; **FileAtt** è una stringa che viene utilizzata per ricostruire il nome completo di un file o di una cartella, mettendo assieme il contenuto di **PercAtt** e **NomeFile**. L'ultima variabile è **DirNum** e servirà da indice per estrarre gli elementi dalla Collection.

```
12. ListaFiles.Clear
13. If Right(PercSorg, 1) <> "\" Then PercSorg = PercSorg & "\"
14. If Dir(PercSorg, TrovaTutto) <> "" Then ListaDir.Add PercSorg
```

All'inizio della funzione (riga 12) viene azzerato il contenuto della *ListBox* di nome **ListaFiles**, per permettere il reinserimento degli elementi.

Alla riga 13 viene controllato se la stringa **PercSorg** termina con un "\"; in caso contrario, la stessa istruzione provvede ad aggiungerlo alla fine della stringa.

La riga 14 effettua un primo controllo: se la cartella di partenza non è vuota, ovvero non contiene né files né sottocartelle, viene aggiunta all'elenco delle cartelle da analizzare in seguito.

Per effettuare la raccolta delle cartelle viene utilizzata la Collection **ListaDir** e la variabile **DirNum** che effettua da indice. Il meccanismo funziona così: viene estratto il primo elemento dalla Collection, che è stato inserito tramite la riga 14. Per tale cartella, vengono ricercate tutte le sottocartelle in essa. Ogni sottocartella trovata viene inserita in coda alla Collection. Al termine della scansione di una cartella, viene incrementato il valore dell'indice **DirNum**, ed il ciclo ricomincia. In tal modo verrà effettuata la medesima operazione per la seconda cartella trovata e per tutte le seguenti, fino a trovare tutte le sottocartelle ed inserirle una per una all'interno della Collection **ListaDir**.

```
15. If InSottoDir Then
16.     DirNum = 1
17.     Do Until DirNum > ListaDir.Count
18.         PercAtt = ListaDir.Item(DirNum)
19.         NomeFile = Dir(PercAtt, TrovaTutto)
20.         Do Until NomeFile = ""
21.             If Left(NomeFile, 1) <> "." Then
22.                 FileAtt = PercAtt & NomeFile
23.                 If (GetAttr(FileAtt) And vbDirectory) Then ListaDir.Add FileAtt & "\"
24.             End If
25.             NomeFile = Dir
26.         Loop
27.         DirNum = DirNum + 1
28.     Loop
29. End If
```

Alla riga 15 inizia un pezzo di codice che termina alla riga 29. Se il parametro **InSottoDir** è True e, quindi, è richiesta l'analisi delle sottocartelle, viene eseguito tale pezzo di codice che effettua la raccolta di tutte le cartelle contenute all'interno della cartella di partenza.

Alla riga 16 viene inizializzata la variabile indice **DirNum** a 1 e alla riga 17 inizia un ciclo che durerà fino a quando il valore dell'indice non sarà maggiore del numero di elementi contenuti in **ListaDir**.

All'interno del ciclo, alla riga 17, viene estratto l'elemento della Collection puntato dall'indice e viene memorizzato nella variabile **PercAtt**. Per tale elemento estratto verranno eseguite le righe dalla 19 alla 26, che elencano le cartelle contenute all'interno della cartella contenuta in **PercAtt**.

Alla riga 19 viene ricercato il primo elemento contenuto nella cartella in analisi. Il nome del primo elemento ricercato viene memorizzato all'interno della variabile **NomeFile**.

Alla riga 20 viene verificato, se l'elemento estratto dalla cartella esista - ovvero il suo nome sia diverso da "". Assicurati di questo, viene effettuato un ulteriore controllo: se il primo carattere dell'elemento estratto è un punto, l'elemento viene scartato. Ciò serve per evitare di elencare le cartelle . e .. che servono soltanto come dispositivi di ritorno.

Alla riga 22 viene costruita la stringa **FileAtt** concatenando le stringhe **PercAtt** e **NomeFile**. In seguito a questo viene controllato se l'elemento in questione sia una cartella, tramite un'[estrazione bit a bit](#). In caso positivo, la cartella viene aggiunta all'elenco delle cartelle **ListaDir**.

Alla riga 25 viene ricercato il prossimo elemento all'interno della cartella **PercAtt** e viene rieseguito il ciclo dalla riga 20, fino a quando non ci sono altri elementi all'interno della cartella.

La prima estrazione ha riportato tutte le sottocartelle contenute all'interno del primo elemento della Collection. Alla riga 27 viene incrementato il valore dell'indice **DirNum** ed il programma ritorna alla riga 17. Viene estratto un ulteriore elemento dalla Collection, vengono elencate tutte le cartelle all'interno d'esso, ed il ciclo continua fino a quando non sono state trovate tutte le cartelle, ovvero fino a quando il valore dell'indice non è maggiore del numero di elementi della Collection.

```
30.   For DirNum = 1 To ListaDir.Count
31.       PercAtt = ListaDir.Item(DirNum)
32.       NomeFile = Dir(PercAtt & TipoFile, TrovaFiles)
33.       Do Until NomeFile = ""
34.           ListaFiles.AddItem NomeFile
35.           NomeFile = Dir
36.       Loop
37.   Next DirNum
38. End Sub
```

Così, alla riga 30 viene reinizializzata la variabile indice **DirNum**, in modo da ricominciare l'estrazione dalla Collection. Stavolta però, non saranno cercate le cartelle, ma soltanto i files all'interno delle singole cartelle contenute nella Collection. Alla riga 31 viene estratto periodicamente un elemento dalla Collection, tramite l'indice **DirNum**, e viene memorizzato nella variabile **PercAtt**.

Per tale cartella, viene ricercato il primo file - costante **TrovaFiles** - contenuto in essa purché rispondente al criterio **TipoFile** (riga 32). Se tale file esiste, vengono eseguite le righe 34 e 35, altrimenti viene estratta un'altra cartella dalla Collection.

Alla riga 34 viene aggiunto il file trovato alla *ListBox*  **ListaFiles**. Solo allora verrà ricercato un altro file contenuto nella medesima cartella, fino ad elencare tutti i files dello stesso criterio per tale cartella.

Terminata una cartella si passerà alla successiva, contenuta nella Collection, fino ad elencare tutti i files del criterio specificato in tutte le cartelle trovate.

Per provare questa funzione scriviamo queste semplici istruzioni:

```
40. Private Sub Elenca_Click()  
41.     Dim TempoInizio As Single  
42.     Dim TempoFine As Single  
43.     Elenca.MousePointer = vbHourglass  
44.     TempoInizio = Timer  
45.     ElencaFiles CercaText.Text, TipoText.Text, SottoCartelle.Value  
46.     TempoFine = Timer  
47.     Tempo.Caption = "Tempo: " & TempoFine - TempoInizio  
48.     Conteggio.Caption = "Conteggio: " & ListaFiles.ListCount  
49.     Elenca.MousePointer = vbNormal  
50. End Sub
```

La funzione è quella legata all'evento Click sul pulsante di nome **Elenca**. Sono state dichiarate 2 variabili di [tipo Single](#), di nome **TempoInizio** e **TempoFine**. Esse serviranno per monitorare la velocità di esecuzione della funzione.

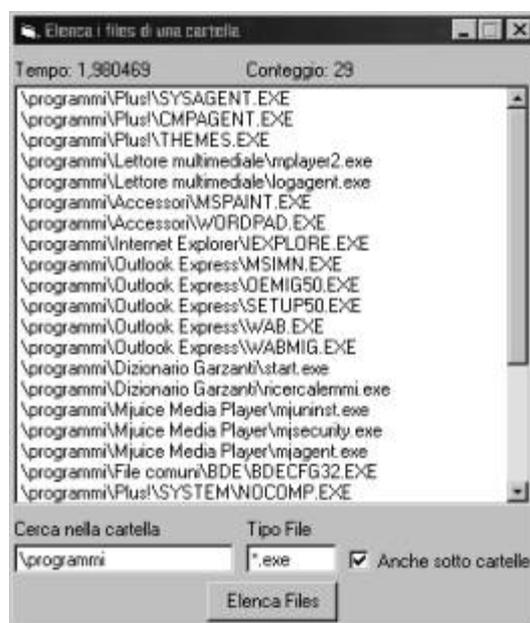
Alla riga 43, viene cambiato il puntatore del mouse in clessidra, poiché tale operazione potrebbe durare alcuni secondi.

Alla riga 44 viene memorizzato l'orario corrente nella variabile **TempoInizio**, viene effettuata la chiamata alla funzione **ElencaFiles** e al termine viene, memorizzato l'orario corrente nella variabile **TempoFine**.

Alla riga 47 viene mostrato nella caption della Label **Tempo** il numero di secondi trascorsi per la scansione delle cartelle. Il valore è dato dalla sottrazione dell'orario iniziale dall'orario finale. Viene anche mostrato il numero di files trovati (riga 48).

Alla riga 49, invece, viene ripristinato il cursore del mouse normale.

Adesso possiamo provare la nostra funzione: il click sopra il pulsante elenca mostra chiaramente i nomi dei files del tipo richiesto (nel nostro caso .exe) contenuti nella cartella specificata (\programmi) e nelle sue sottocartelle.



## Figura 2

[Segue parte 2 >>](#)

[Fibia FBI](#) e [Giuseppe Della Bianca](#)  
27 Dicembre 2000



[Torna all'indice degli HowTo](#)

---