


[Home Page](#)
[Novità](#)
[Aiuto](#)

Eseguire operazioni di Shell sui files

http://www.vbsimple.net/howto/ht_056.htm
Difficoltà: 2 / 5

Quasi tutti i moderni programmi affidano alle funzioni di Shell il compito delle operazioni di copia, spostamento ed eliminazione dei files. In tal modo infatti il programmatore si scarica di alcuni compiti di analisi, in particolare la reiterazione per le sottocartelle. L'utente invece si trova a contatto con un'interfaccia grafica conosciuta ed apprezzata. Vedremo in questo progetto come richiamare le funzioni della shell per queste operazioni.

La classe *clsFBIShellFileOperation* sviluppata in questo progetto consentirà di eseguire le operazioni più comuni con estrema semplicità:

```

1. Option Explicit
2. Option Base 0
3.
4. Public Enum SHOperations
5.     FO_MOVE = &H1
6.     FO_COPY = &H2
7.     FO_DELETE = &H3
8.     FO_RENAME = &H4
9. End Enum
10.
```

L'[enumerazione](#) *SHOperations* si compone di 4 costanti definite dall'[API](#) che specificano il tipo di operazione da svolgere tra spostamento, copia, cancellazione e rinominazione. Sarà utilizzata più avanti dalla struttura **SHFILEOPSTRUCT**.

```

11. Public Enum SHOperationFlags
12.     FOF_MULTIDESTFILES = &H1
13.     FOF_CONFIRMMOUSE = &H2
14.     FOF_SILENT = &H4
15.     FOF_RENAMEONCOLLISION = &H8
16.     FOF_NOCONFIRMATION = &H10
17.     FOF_WANTMAPPINGHANDLE = &H20
18.     FOF_ALLOWUNDO = &H40
19.     FOF_FILESONLY = &H80
20.     FOF_SIMPLEPROGRESS = &H100
21.     FOF_NOCONFIRMMKDIR = &H200
22.     FOF_NOERRORUI = &H400
23.     FOF_NOCOPYSECURITYATTRIBS = &H800
24.     FOF_NORECURSION = &H1000
25.     FOF_NO_CONNECTED_ELEMENTS = &H2000
26.     FOF_WANTNUKEWARNING = &H4000
27. End Enum
28.
```

L'enumerazione *SHOperationFlags* consente di specificare uno o più comportamenti aggiuntivi durante l'operazione di Shell. I loro nomi dovrebbero da soli essere chiari ma merita una piccola citazione il flag **FOF_ALLOWUNDO** (riga 18) che consente di annullare l'ultima operazione eseguita, mediante la voce *Annulla* del menu popup in una qualunque cartella. Funziona regolarmente per le operazioni di copia e spostamento, ma nel caso dell'operazione di cancellazione, differenzia lo spostamento nel cestino

dall'eliminazione vera e propria; la specifica del flag **FOF_ALLOWUNDO** nell'operazione di cancellazione comporta quindi lo spostamento nel cestino dei files.

```

29. Private Type SHFILEOPSTRUCT
30.     hwnd As Long
31.     wFunc As Long
32.     pFrom As String
33.     pTo As String
34.     fFlags As Long
35.     fAnyOperationsAborted As Long
36.     hNameMappings As Long
37.     lpzProgressTitle As String
38. End Type
39.
```

La struttura **SHFILEOPSTRUCT**, riportata in maniera errata dalla maggior parte delle documentazioni, è stata trattata in [un articolo della sezione Weird Things](#), è stata corretta per consentire un corretto funzionamento della funzione in tutte le situazioni. Il campo **wFunc** della struttura specifica il tipo di operazione da eseguire e determina l'uso degli altri campi; ad esempio durante la copia di files il campo **pFrom** indicherà l'origine dei files da copiare, mentre **pTo** conterrà la destinazione per i files; nel caso invece della cancellazione, il campo **pTo** non dovrà essere specificato. I valori ammessi per il campo **wFunc** sono quelli contenuti nell'enumerazione **SHOperations**, mentre per il campo **fFlags** sarà necessario specificare uno o più valori dell'enumerazione **SHOperationFlags**.

```

40. Private Declare Function SHFileOperation Lib "SHELL32.DLL" Alias
    "SHFileOperationA" (lpFileOp As SHFILEOPSTRUCT) As Long
41.
42. Private udtOperation As SHFILEOPSTRUCT
43.
```

L'intero codice è basato sull'uso della funzione *SHFileOperation* che richiede la specifica di una variabile di tipo **SHFILEOPSTRUCT**, definito in precedenza; tale valore è dichiarato alla riga 42, col nome **udtOperation**.

```

44. Private Sub Class_Initialize()
45.     udtOperation.fFlags = FOF_NOERRORUI
46. End Sub
47.
```

Il valore predefinito per il campo **fFlags** sarà **FOF_NOERRORUI**, che specifica di non mostrare le videate di errore della Shell, ma lasciare il compito al programma.

```

48. Public Property Get AllowUndo() As Boolean
49.     AllowUndo = CBool(udtOperation.fFlags And FOF_ALLOWUNDO)
50. End Property
51.
52. Public Property Let AllowUndo(ByVal newValue As Boolean)
53.     With udtOperation
54.         .fFlags = IIf(newValue, .fFlags Or FOF_ALLOWUNDO, .fFlags And Not
            FOF_ALLOWUNDO)
55.     End With
56. End Property
57.
```

La proprietà **AllowUndo** è utilizzata per recuperare ed assegnare con semplicità il flag **FOF_ALLOWUNDO** accennato in precedenza e consentire l'annullamento dell'ultima operazione eseguita.

```

58. Public Property Get Confirm() As Boolean
59.     Confirm = Not CBool(udtOperation.fFlags And FOF_NOCONFIRMATION)
60. End Property
61.
62. Public Property Let Confirm(ByVal newValue As Boolean)
63.     With udtOperation
64.         .fFlags = IIf(newValue, .fFlags And Not FOF_NOCONFIRMATION, .fFlags Or
FOF_NOCONFIRMATION)
65.     End With
66. End Property
67.

```

La proprietà **Confirm** consente di richiedere conferma all'utente prima di eseguire certe operazioni di sovrascrittura o cancellazione. Si tratta in sostanza di recuperare o assegnare il valore del flag **FOF_NOCONFIRMATION**.

```

68. Public Property Get Destination() As String
69.     Destination = udtOperation.pTo
70. End Property
71.
72. Public Property Let Destination(ByVal newValue As String)
73.     udtOperation.pTo = newValue
74. End Property
75.
76. Public Property Get Flags() As SHOperationFlags
77.     Flags = udtOperation.fFlags
78. End Property
79.
80. Public Property Let Flags(ByVal newValue As SHOperationFlags)
81.     udtOperation.fFlags = newValue
82. End Property
83.
84. Public Property Get Source() As String
85.     Source = udtOperation.pFrom
86. End Property
87.
88. Public Property Let Source(ByVal newValue As String)
89.     udtOperation.pFrom = newValue
90. End Property
91.


```



Le tre proprietà **Destination**, **Flags** e **Source** consentono di recuperare o assegnare rispettivamente la destinazione (**pTo**), i flags (**fFlags**) e l'origine (**pFrom**) della variabile **udtOperation** che verrà utilizzata per eseguire l'operazione di Shell.

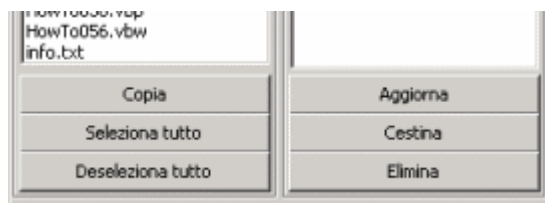
```

92. Public Function Esegui(ByVal Operation As SHOperations) As Long
93.     udtOperation.wFunc = Operation
94.     Esegui = SHFileOperation(udtOperation)
95. End Function

```

L'unico metodo  della classe *clsFBIShellFileOperation* è **Esegui** e richiede la specifica del tipo di operazione da eseguire, corrispondente ad un valore dell'enumerazione **SHOperations**. Il valore sarà assegnato al membro **wFunc** della struttura **udtOperation** e quindi sarà richiamata la funzione *SHFileOperation* con la stessa struttura.

Svilupperemo una semplice interfaccia per il nostro progetto, formata ad un controllo *DirListBox*  di nome **dirOrigine** ed un controllo *FileListBox*  collegato di nome **filOrigine**. In fondo a queste



avremo tre pulsanti di nome **cmdCopia**, **cmdSelezionaTutto** e **cmdDeselezionaTutto**. Sul lato destro avremo invece un altro controllo *FileListBox* di nome **filTemp** che punterà sempre alla stessa cartella del programma; anche in fondo a questo esistono tre pulsanti di comando, di nome **cmdAggiorna**, **cmdCestina** e **cmdElimina**. Il controllo **filOrigine** avrà inoltre la proprietà *MultiSelect* impostata su **1 - Simple**.

Come già detto, i due controlli **dirOrigine** e **filOrigine** saranno collegati, per cui al cambio della cartella sul controllo superiore corrisponderà il cambio della cartella nel secondo. Vediamo il semplice codice che gestisce tutti gli eventi del form:

```
1. Option Explicit
2.
3. Private fbiShellOperation As clsFBIShellFileOperation
4.
```

Alla riga 3 è dichiarata un'[istanza](#) della classe *clsFBIShellFileOperation*, che verrà dichiarata in occasione del caricamento del form.

```
5. Private Function LastBSlash(ByVal Path As String) As String
6.     If Right$(Path, 1) <> "\" Then Path = Path & "\"
7.     LastBSlash = Path
8. End Function
9.
10. Private Sub Form_Load()
11.     Set fbiShellOperation = New clsFBIShellFileOperation
12.     filTemp.Path = LastBSlash(App.Path) & "TEMP"
13. End Sub
14.
15. Private Sub Form_Unload(Cancel As Integer)
16.     Set fbiShellOperation = Nothing
17. End Sub
18.
```

La funzione **LastBSlash**, trattata anche in altri articoli del sito, è utilizzata per assicurare che il percorso fornito termini con una barra rovesciata (\) e poterlo quindi concatenare ad un nome di file ed avere quindi la sicurezza che il percorso fornito sia valido. Prendiamo ad esempio il percorso C:\ da concatenare ad un nome di file quale "file1.txt"; nel caso del percorso "C:\" non sarà necessario aggiungere la barra rovesciata ("C:\" & "file1.txt"), ma nel caso di un altro percorso quale "C:\Cartella", la stessa sarà mancante e dovrà essere specificata prima del nome del file. La funzione **LastBSlash** restituisce sempre e comunque un percorso che termini con la barra rovesciata, da concatenare quindi al nome di un file.

La funzione sarà utilizzata alla riga 12, al caricamento del form, per inizializzare il controllo **filTemp** e farlo puntare alla cartella **TEMP** contenuta nella cartella del nostro programma; in occasione dell'operazione, sarà anche istanziata la variabile **fbiShellOperation**, che verrà quindi [deallocata](#) alla chiusura del form.

```
19. Private Sub dirOrigine_Change()
20.     filOrigine.Path = dirOrigine.Path
21. End Sub
22.
23. Private Sub cmdAggiorna_Click()
24.     filTemp.Refresh
25. End Sub
26.
```

Come già detto, al cambio della cartella nel controllo **dirOrigine** corrisponderà il cambio della cartella del controllo **filOrigine** e ciò è gestito alle righe 19-21.

La pressione del pulsante **cmdAggiorna** invece causerà l'aggiornamento dell'altro elenco di files, di nome **filTemp**.

```
27. Private Sub cmdSelezionaTutto_Click()  
28.     Dim intLoop As Integer  
29.     For intLoop = 0 To filOrigine.ListCount - 1  
30.         filOrigine.Selected(intLoop) = True  
31.     Next intLoop  
32. End Sub  
33.  
34. Private Sub cmdDeselezionaTutto_Click()  
35.     Dim intLoop As Integer  
36.     For intLoop = 0 To filOrigine.ListCount - 1  
37.         filOrigine.Selected(intLoop) = False  
38.     Next intLoop  
39. End Sub  
40.
```

Le due routine legate al click dei pulsanti **cmdSelezionaTutto** e **cmdDeselezionaTutto** sono molto simili e comprendono un semplice ciclo per ogni elemento dell'elenco **filOrigine**. La prima routine assegnerà lo stato di selezionato ad ogni elemento, mentre la seconda deselezionerà tale stato.

```
41. Private Sub cmdCopia_Click()  
42.     Dim intLoop As Integer  
43.     Dim strFiles As String  
44.     With filOrigine  
45.         For intLoop = 0 To .ListCount - 1  
46.             If .Selected(intLoop) Then strFiles = strFiles & LastBSlash  
              (filOrigine.Path) & .List(intLoop) & vbNullChar  
47.         Next intLoop  
48.         If Len(strFiles) > 0 Then  
49.             strFiles = strFiles & vbNullChar  
50.             With fbiShellOperation  
51.                 .Confirm = True  
52.                 .AllowUndo = False  
53.                 .Source = strFiles  
54.                 .Destination = filTemp.Path  
55.                 Call .Esegui(FO_COPY)  
56.             End With  
57.             cmdAggiorna_Click  
58.         End If  
59.     End With  
60. End Sub  
61.
```

La prima routine che utilizza l'istanza **fbiShellOperation** è quella legata all'operazione di copia selezionati nell'elenco **filOrigine**. Sarà pertanto lanciato un ciclo per tutti gli elementi dell'elenco (riga 46) ed il nome (completo di percorso) di ciascun file selezionato sarà concatenato alla variabile **strFiles**. I nomi saranno tra loro separati da un carattere NULL.

Se almeno un file sarà stato accodato alla variabile **strFiles**, sarà quindi aggiunto un ulteriore carattere terminatore (riga 49) e quindi invocata la procedura di copia dei files selezionati. Saranno impostate le proprietà **Confirm** su True, **AllowUndo** su False, **Source** sull'elenco di files da copiare e **Destination** sul percorso della cartella temporanea (indicato dalla proprietà *Path* del controllo **filTemp**, inizializzato all'avvio del programma). Sarà quindi lanciata la procedura di copia mediante metodo **Esegui**.

Al termine dell'operazione sarà forzato l'aggiornamento dell'elenco **filTemp**, richiamando la routine **cmdAggiorna_Click**.

```

62. Private Sub cmdCestina_Click()
63.     fbiShellOperation.AllowUndo = True
64.     EliminaFiles
65. End Sub
66.
67. Private Sub cmdElimina_Click()
68.     fbiShellOperation.AllowUndo = False
69.     EliminaFiles
70. End Sub
71.

```

Le due routine di eliminazione definitiva o provvisoria, si differenziano soltanto per l'assegnazione della proprietà **AllowUndo** dell'istanza **fbiShellOperation**. Entrambe affidano l'operazione alla routine **EliminaFiles** che vediamo qui di seguito:

```

72. Private Sub EliminaFiles()
73.     With fbiShellOperation
74.         .Source = LastBSlash(filTemp.Path) & ".*"
75.         .Destination = ""
76.         .Confirm = True
77.         .Esegui FO_DELETE
78.     End With
79.     cmdAggiorna_Click
80. End Sub

```

La routine assegna come origine dei files da eliminare il percorso della cartella TEMP, concatenato con i [caratteri jolly](#) *.*; la destinazione dovrà essere nulla, sarà attivata la conferma (riga 76) e lanciata quindi l'operazione alla riga successiva. Al termine dell'operazione sarà richiesto l'aggiornamento dell'elenco **filTemp** mediante richiamo della routine **cmdAggiorna_Click**.

Durante le operazioni di copia, spostamento nel cestino ed eliminazione dei files saranno mostrate le finestre di dialogo standard in ambiente Windows, analoghe a quelle mostrate a fianco.

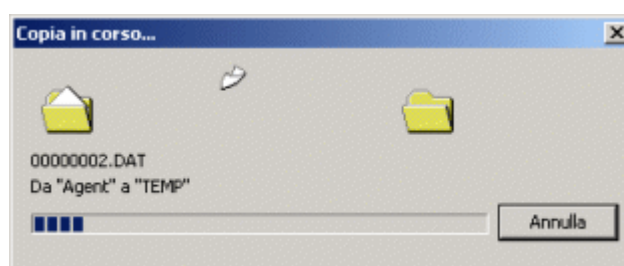


Figura 2

Resta tuttavia impossibile modificare il titolo mostrato sulla barra delle suddette finestre di dialogo ed il testo in esso contenuto.

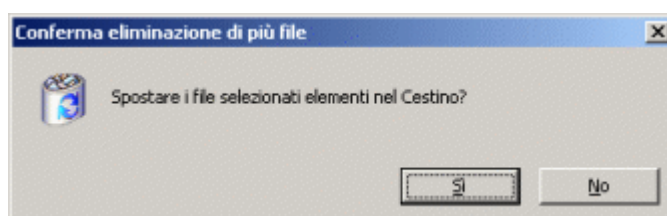


Figura 3

Ad esempio durante l'operazione di spostamento nel cestino di uno o più files sarà mostrato il testo *"Spostare i file selezionati elementi nel Cestino?"*, mentre nel caso di eliminazione diretta apparirà il testo *"Eliminare tutti gli elementi presenti nel Cestino?"*, entrambi per nulla chiari, anzi fuorvianti

con le azioni da svolgere.

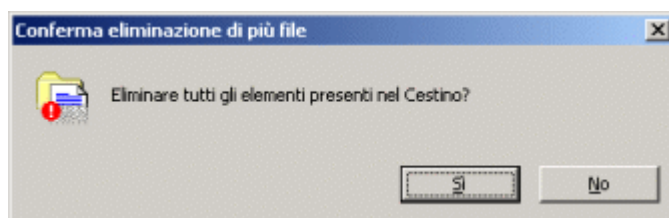


Figura 4

La classe *clsFBIShellFileOperation* consente una minima gestione delle operazioni sui files con estrema semplicità e dovrebbe essere preferita alla copia manuale di uno o più files, in modo da rendere l'ambiente in cui si opera quanto più familiare possibile con il look dei vari Windows in cui il programma viene eseguito.

[Fibia FBI](#)
3 Giugno 2003



[Torna all'indice degli HowTo](#)
