



Associare un programma ad un'estensione (seconda parte)

http://www.vbsimple.net/howto/ht_054_2.htm

Difficoltà:  5 / 5

[<< Continua dalla parte 1](#)

Segue l'unico metodo pubblico della classe: **Register**, che consente di registrare nel sistema l'estensione specificata ed associarvi un verbo, completo di descrizione aggiuntiva e del percorso del programma da richiamare.

```

106. Public Function Register(ByVal newName As String, ByVal newFullName As String,
    ByVal newCommandPath As String) As Boolean
107.     Dim fbiRegistry As clsFBIRegistry
108.     Dim strNewPath As String
109.     Register = False
110.     m_SavedPath = ""
111.     If ShowError(Len(m_strExt) = 0, "Specificare un valore per la proprietà
        Extension") Then Exit Function
112.     If ShowError(Len(m_strFileType) = 0, "Specificare un valore per la proprietà
        FileType") Then Exit Function
113.     If ShowError(Len(newName) = 0, "Specificare un valore per il nome del verbo")
        Then Exit Function
114.     Set fbiRegistry = New clsFBIRegistry
115.
```

Alla riga 107 è dichiarata la variabile **fbiRegistry** che verrà utilizzata per accedere al registro mediante la classe *clsFBIRegistry*. Alla riga 114 è istanziata la variabile **fbiRegistry**. L'altra variabile **strNewPath** sarà utilizzata per determinare la posizione in cui andranno memorizzate le impostazioni.

La funzione restituisce un valore booleano corrispondente all'avvenuta registrazione. Durante l'avvio saranno verificati i valori delle proprietà **Extension**, **FileType** e del verbo da registrare. In caso di valori nulli sarà generato un errore e la funzione sarà interrotta.

```

116.     With fbiRegistry
117.         Call .CreaChiave(HKEY_CLASSES_ROOT, "." & m_strExt)
118.         If m_PreserveSettings Then
119.             strNewPath = .Valore("") & ""
120.             If Len(strNewPath) > 0 Then
121.                 m_DeferSettings = True
122.                 Call .CreaChiave(HKEY_CLASSES_ROOT, strNewPath)

```

Poiché verrà utilizzata dappertutto l'istanza **fbiRegistry** si è scelto di attivare il qualificatore immediato con **With**. La prima operazione da effettuare è quella di aprire (o creare se inesistente) la chiave corrispondente all'estensione da registrare.

La riga 118 spezza la funzione in due grosse parti: se la proprietà **PreserveSettings** è impostata su **True** allora dovranno essere rilevate le impostazioni attuali e preservate; in caso contrario dovranno essere utilizzate le altre proprietà della classe, senza alcun rispetto dei valori attuali.

Nel caso si scelga di preservare le impostazioni attuali, la prima operazione da eseguire sarà quella di capire se attualmente è utilizzato il deferring e tale verifica è fatta alle righe 119-120. Se esiste un valore di deferring sarà ridefinito il valore della proprietà **DeferSettings** e sarà aperta la chiave puntata dal valore di deferring.

```

123.      Else
124.      If .ApriChiave(CHIAVE_APERTA, "shell") <> 0 Then
125.      m_DeferSettings = False
126.      strNewPath = "." & m_strExt
127.      Else
128.      If m_DeferSettings Then
129.      strNewPath = m_DeferredPath
130.      Call .ApriChiave(HKEY_CLASSES_ROOT, "." & m_strExt)
131.      .Valore("", REG_SZ) = strNewPath
132.      Else
133.      strNewPath = "." & m_strExt
134.      End If
135.      End If
136.      Call .ChiudiChiave
137.      Call .CreaChiave(HKEY_CLASSES_ROOT, strNewPath)
138.      If (Len(m_strFileType) > 0) And (m_DeferSettings) Then .Valore("", REG_SZ)
= m_strFileType
139.      End If

```

Le cose si complicano se la registrazione attuale non utilizza deferring. Ciò infatti può significare che la chiave non esiste oppure esiste ma semplicemente non fa uso di una registrazione reinviata.

Il primo controllo quindi da effettuare consiste nel verificare che esista una sottochiave di nome *shell* (riga 124). In tal caso sarà ridefinito il valore della proprietà **DeferSettings** per informare l'utente che le impostazioni attuali non utilizzano deferring e ci si prepara a lavorare con una chiave non rinviata.

Nel caso che la sottochiave shell non esista invece il programma dovrà comportarsi secondo le impostazioni delle proprietà **DeferSettings** e **DeferredPath**. Alla riga 131 sarà assegnato il valore per la chiave reinviata che sarà creata alla riga 137.

Se la proprietà **FileType** contiene qualche valore e si sta applicando il deferring allora sarà aggiunta anche la descrizione aggiuntiva del tipo di file.

```

140.      m_SavedPath = "HKEY_CLASSES_ROOT\" & strNewPath
141.      Call .CreaChiave(CHIAVE_APERTA, "shell")
142.      If Len(m_strDefaultVerb) > 0 Then .Valore("") = m_strDefaultVerb
143.      Call .CreaChiave(CHIAVE_APERTA, newName & "\command")
144.      If ShowError(Len(.Valore("", REG_SZ) & "") > 0, "Impossibile sovrascrivere il
verbo " & UCCase$(newName) & vbNewLine & "se la proprietà PreserveSettings è
impostata su True") Then
145.      Call .ChiudiChiave
146.      Set fbiRegistry = Nothing
147.      Exit Function
148.      Else
149.      Call .ChiudiChiave
150.      Call .ApriChiave(HKEY_CLASSES_ROOT, strNewPath)
151.      If Len(m_strDefaultIcon) > 0 Then
152.      Call .CreaChiave(CHIAVE_APERTA, "DefaultIcon")
153.      .Valore("", REG_SZ) = m_strDefaultIcon
154.      Call .ChiudiChiave
155.      End If
156.      Call .ApriChiave(HKEY_CLASSES_ROOT, strNewPath & "\shell\" & newName)
157.      End If

```

Questa è la parte più complessa del codice. Dopo aver registrato il valore della chiave in cui andranno memorizzati i verbi viene creata la sottochiave *shell* (riga 141). Se è stato assegnato un valore alla proprietà **DefaultVerb** questo sarà immediatamente scritto nel valore predefinito della chiave appena creata.

Saranno quindi create con una sola riga le due sottochiavi corrispondenti al verbo richiesto nella registrazione (**newName**) e la corrispondente sottochiave *command*. Se la chiave esistesse già e ci fosse un valore predefinito l'elaborazione non dovrà procedere ulteriormente per non sovrascrivere i dati preesistenti. Infatti la proprietà **PreserveSettings** non consente di sovrascrivere verbi esistenti. Saranno quindi in ordine generato l'evento **Error**, chiusa l'ultima chiave usata, deallocata la variabile **fbiRegistry** ed interrotta l'esecuzione della funzione (righe 144-147).

Se invece la chiave non esistesse sarà semplicemente chiusa la chiave *command* aperta in precedenza ed aperta la chiave corrispondente al punto di registrazione (estensione o deferring). Dovrà quindi essere registrata l'icona specificata nella proprietà **DefaultIcon** (righe 151-155).

Completate queste premesse ci si potrà preparare alla registrazione del verbo vero e proprio.

```

158.     Else
159.         If m_DeferSettings Then
160.             .Valore("", REG_SZ) = m_DeferredPath
161.             Call .CreaChiave(HKEY_CLASSES_ROOT, m_DeferredPath, False)
162.             m_SavedPath = "HKEY_CLASSES_ROOT\" & m_DeferredPath
163.             If Len(m_strFileType) > 0 Then .Valore("", REG_SZ) = m_strFileType
164.         Else
165.             Call .EliminaValore("")
166.             m_SavedPath = "HKEY_CLASSES_ROOT\" & m_strExt
167.         End If
168.         If Len(m_strDefaultIcon) > 0 Then
169.             Call .CreaChiave(CHIAVE_APERTA, "DefaultIcon")
170.             .Valore("", REG_SZ) = m_strDefaultIcon
171.             Call .ChiudiChiave
172.             Call .ApriChiave(HKEY_CLASSES_ROOT, IIf(m_DeferSettings, m_DeferredPath,
173. " " & m_strExt))
174.             End If
175.             Call .CreaChiave(CHIAVE_APERTA, "shell")
176.             If Len(m_strDefaultVerb) > 0 Then .Valore("") = m_strDefaultVerb
177.             Call .CreaChiave(CHIAVE_APERTA, newName)
178.         End If

```

Resta da valutare l'intera situazione in cui è stato scelto di non preservare le impostazioni attuali. Sarà verificato se dalla proprietà **DeferSettings** è stato scelto di usare il deferring: in tal caso sarà applicato immediatamente registrandone il valore e creando la nuova chiave (righe 159-162). Sarà inoltre aggiunta la descrizione completa del tipo di file (riga 163).

Nel caso non si desiderasse utilizzare il deferring è fondamentale assicurarsi che sia eliminato l'eventuale deferring preesistente, che altrimenti avrebbe la priorità (riga 165).

Se è stata impostata un'icona nella proprietà **DefaultIcon** essa sarà registrata alle righe 168-173. Lo stesso vale per il verbo di default (riga 175).

Al raggiungimento della riga 178 la chiave correntemente aperta punterà esattamente al verbo da registrare (righe 156 e 176).

```
178.      If Len(newFullName) > 0 Then .Valore("", REG_SZ) = newFullName
179.      Call .CreaChiave(CHIAVE_APERTA, "command")
180.      .Valore("", REG_SZ) = newCommandPath
181.      Call .ChiudiChiave
182.  End With
183.  Set fbiRegistry = Nothing
184.  Register = True
185. End Function
```

Segue la fase conclusiva della registrazione: la scrittura della descrizione aggiuntiva al verbo, se esiste nel parametro **newFullName**. Sarà infine creata la chiave **command** in cui sarà scritto il percorso del programma da richiamare per il verbo indicato.

La routine volge al termine con la chiusura dell'ultima chiave usata, la [deallocazione](#) della variabile **fbiRegistry** e l'impostazione del valore di ritorno a True, ad indicare la corretta registrazione del verbo.

Nell'esempio allegato è presente anche un progetto che utilizza questa classe con dati dimostrativi ma fondalmente inutili. Saranno utilizzate tutte le proprietà e le due modalità di registrazione dei dati, mediante scrittura sulla chiave dell'estensione e mediante deferring.

Si raccomanda di guardare i cambiamenti direttamente sul registro di Windows man mano che vengono chiamate le funzioni **Register**.

[Fibia FBI](#)

22 Dicembre 2002



[Torna all'indice degli HowTo](#)
