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

Generare dei files in base ad un particolare tracciato (terza parte)

http://www.vbsimple.net/activity/act_11_3.htm

Richiesta di: [Roberto Corona](#) - 12 Gennaio 2001

Difficoltà:  5 / 5

Ho bisogno di scrivere un programmino che, dopo aver analizzato dei files contenenti dei particolari tracciati, estraiga dei dati da un foglio di Excel e tramite questi generi dei files di testo con una particolare struttura determinata dal file del tracciato.

[<< Continua dalla parte 2](#)

Terminata la visione e lo studio del modulo standard del progetto possiamo passare ai due form che compongono l'interfaccia utente.

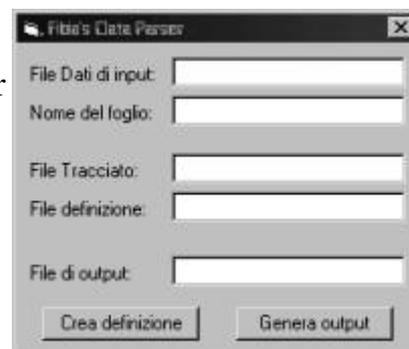
Poiché i due form sono strettamente connessi, nello studio sarà necessario dare un primo sguardo al primo form, studiare il secondo e una volta fatto, terminare lo studio del primo form.

Cominciamo con la semplicissima interfaccia utente del form principale.

Essa consiste in 5 *Label* **A** e 5 *TextBox* **abl** e in basso appaiono due semplici *CommandButton* **■** che serviranno per accedere al secondo form e confermare i dati immessi.

Abbiamo quindi nell'ordine:

- Una *Label* **A** di nome **DatiInputLabel** con la *Caption* impostata a "**File Dati di Input:**";
- Una *TextBox* **abl** di nome **DatiInputText**, che dovrà contenere il nome del file di Excel contenente i dati da processare;
- Una *Label* **A** di nome **NomeFoglioLabel** con la *Caption* impostata a "**Nome del foglio:**";
- Una *TextBox* **abl** di nome **NomeFoglioText** che dovrà contenere il nome del foglio del file di Excel con i dati da processare;
- Una *Label* **A** di nome **TracciatoLabel** con la *Caption* impostata a "**File Tracciato:**";
- Una *TextBox* **abl** di nome **TracciatoText** che dovrà contenere il nome del file di testo del tracciato;
- Una *Label* **A** di nome **DefinizioneLabel** con la *Caption* impostata a "**File definizione:**";



- Una *TextBox*  di nome **DefinizioneText** che dovrà contenere il nome di un file definizione .FDF salvato tramite l'utility di Creazione delle definizioni;
- Una *Label* **A** di nome **OutputLabel** con la *Caption* impostata a "**File di Output:**";
- Una *TextBox*  di nome **OutputText** che dovrà contenere il nome del file di testo in cui saranno salvati i dati. Tale file non deve esistere su Hard disk e nel caso che ci fosse non sarà tentata la scrittura;
- Un *CommanButton*  di nome **CreaDefinizione** con la *Caption* impostata a "**Crea definizione**" che servirà per richiamare l'utility di Definizione delle definizioni;
- Un *CommanButton*  di nome **GeneraOutput** con la *Caption* impostata a "**Genera output**" che servirà ad accettare tutti i dati immessi e generare il file .TXT voluto con questo programma.

Il codice è molto semplice: saranno gestiti soltanto tre [eventi](#) ⚡ in questo form: il click sopra il pulsante "**Crea definizione**", il click sopra il pulsante "**Genera output**" e la chiusura del programma per effettuare la chiusura del database *EXCEL* aperto.

Inizialmente vedremo soltanto il primo e l'ultimo evento, poiché ancora non sappiamo come sia costituito il file di definizione.

Ecco quindi le poche righe di codice:

```

1. Option Explicit
2.
3. Private Sub CreaDefinizione_Click()
4.     On Error GoTo ERRORE
5.     If FileEsiste(DatiInputText.Text) = False Then Exit Sub
6.     If FileEsiste(TracciatoText.Text) = False Then Exit Sub
7.     If Len(Trim(NomeFoglioText.Text)) = 0 Then Exit Sub
8.     Set FILEXLS = OpenDatabase(DatiInputText.Text, 0, 0, "Excel 5.0")
9.     Set FOGLIOXLS = FILEXLS.OpenRecordset(NomeFoglioText.Text & "$")
10.    Load DefinitionForm
11.    DefinitionForm.FileTracciato.LoadFile TracciatoText.Text, rtfText
12.    DefinitionForm.Show vbModal
13.    Unload DefinitionForm
14.    Exit Sub
15. ERRORE:
16.    MsgBox "Si è verificato un errore.", vbCritical + vbOKOnly, "Genera output"
17.    Set FOGLIOXLS = Nothing
18.    Set FILEXLS = Nothing
19. End Sub
20.

```

Nel momento in cui l'utente clicca sul pulsante *CreaDefinizione* viene controllato se i dati immessi nelle caselle *DatiInputText*, *TracciatoText* e *NomeFoglioText* sono validi mediante verifica dell'esistenza del file e apertura del foglio di Excel.

Alla riga 4 abbiamo un'istruzione di gestione degli errori; per cui ogni errore farà sì che la Sub giunga al termine.

Alla riga 8 abbiamo l'apertura del database Excel tramite driver [ISAM](#) "Excel 5.0", compatibile con tutte le versioni di Excel. Alla riga successiva viene effettuata l'apertura del foglio specificato nella casella di testo apposita.

Nota che i nomi dei fogli di testo sono memorizzati in una maniera per la quale il nome del

foglio è sempre seguito da un segno di dollaro (\$); ecco perché l'apertura del Recordset ha come parametro il nome del foglio ed il segno di dollaro.

Alla riga 10 viene caricato il form **DefinitionForm**, che si occuperà di generare i file di definizione, ma non viene mostrato subito. Prima viene caricato il file di tracciato all'interno della *RichTextBox*  **FileTracciato** tramite il metodo *LoadFile*. Subito dopo questo viene mostrato il form in modalità [modale](#), ovvero a scelta obbligatoria (riga 12).

Da questo punto in poi la gestione viene affidata completamente al form **DefinitionForm**. Il nostro form principale si occuperà soltanto di chiudere il form all'uscita da esso.

In caso fosse generato qualche errore in questa Sub verrà mostrato un messaggio di errore (riga 16) e verranno chiusi e [deallocati](#) il Recordset ed il database Excel.

```

21. Private Sub Form_Unload(Cancel As Integer)
22.     On Error Resume Next
23.     FOGLIOXLS.Close
24.     FILEXLS.Close
25.     Set FOGLIOXLS = Nothing
26.     Set FILEXLS = Nothing
27. End Sub
28.

```

La chiusura del form si assicura soltanto che vengano chiusi e deallocati correttamente il Recordset FOGLIOXLS ed il database FILEXLS.

Prima di vedere il funzionamento della routine di creazione del file di output è necessario vedere come si compone il file di definizione.

Vediamo prima, quindi, il form che si occuperà della definizione del tracciato. Il nome del form è **DefinitionForm** e si presenta abbastanza complesso.

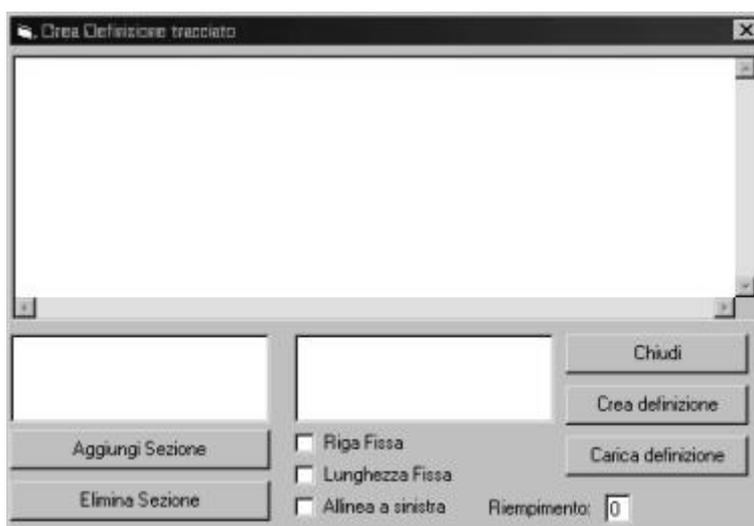


Figura 2

Nella parte superiore del form abbiamo una *RichTextBox*  di nome **FileTracciato** con le proprietà impostate come segue:

- *AutoVerbMenu* = False
- *DisableNoScroll* = True

- *Font* = Courier New normale dimensione 8
- *HideSelection* = False
- *Locked* = True
- *MultiLine* = True
- *RightMargin* = 65000
- *ScrollBars* = 3 - Both

Nella sezione sotto abbiamo due *ListBox*  di nome **ListaSezioni** e **ListaCampi**; abbiamo anche una serie di *CommandButton*  i cui nomi sono: **PulsanteChiudi**, **PulsanteCreaDefinizione**, **PulsanteCaricaDefinizione**, **PulsanteAggiungiSezione**, **PulsanteEliminaSezione**. A questi si aggiungono tre *CheckBox*  di nome **RigaFissaCheck**, **LunghezzaFissaCheck** e **AllineaSinistraCheck**. Completano l'interfaccia una *Label* **A** di nome **FillerLabel** ed una *TextBox*  di nome **FillerText** con la proprietà *MaxLength* impostata a 1.

In sostanza, tramite questo form sarà possibile analizzare il tracciato e definire le posizioni in cui andranno inseriti i dati estratti dal file Excel. Una volta terminata la definizione sarà possibile salvarla in un file FDF. Inoltre sarà possibile caricare files FDF salvati in precedenza.

L'analisi del tracciato si effettua separando in sezioni le parti che devono subire un trattamento particolare, quale la sostituzione con un campo del database.

La *RichTextBox*  **FileTracciato** servirà per selezionare parti del testo e dividerle in sezioni.

Le sezioni saranno gestite tramite le funzioni della parte sinistra, mentre la parte centrale è dedicata alla definizione delle singole sezioni.

Accennato il funzionamento del form **DefinitionForm**, possiamo tuffarci nel complesso codice che ne regola il comportamento.

```
1. Option Explicit
2. Private SEZIONI As Collection
3. Private TROVAPOSIZIONE As Boolean
4.
```

Abbiamo definito due variabili globali all'interno del form. La prima è una *Collection* di nome **SEZIONI** che conterrà tutte le sezioni convertite in stringa tramite la funzione **Sezione2Riga**. La seconda variabile è **TROVAPOSIZIONE**, di tipo booleano e servirà per inibire l'esecuzione di certi eventi nel momento in cui cambieremo alcuni stati.

```
5. Public Sub MostraSezioni()
6.     Dim CONTA As Integer
7.     Dim TMPSEZIONE As TIPOSEZIONE
8.     ListaSezioni.Clear
9.     For CONTA = 1 To SEZIONI.Count
10.        Riga2Sezione SEZIONI.Item(CONTA), TMPSEZIONE
11.        ListaSezioni.AddItem "Da " & TMPSEZIONE.INIZIO & " A " & TMPSEZIONE.FINE
12.     Next CONTA
13.     ListaSezioni.ListIndex = SEZIONI.Count - 1
14. End Sub
15.
```

Prima di addentrarci nel form vediamo la Sub **MostraSezioni**. La sua esecuzione comporta l'azzeramento della *ListBox* **ListaSezioni** (riga 8) ed il successivo riempimento tramite lettura della *Collection* SEZIONI.

```

16. Private Sub Form_Load()
17.     Dim CONTA As Integer
18.     Set SEZIONI = New Collection
19.     ListaCampi.AddItem "(nessuno)"
20.     For CONTA = 0 To FOGLIOXLS.Fields.Count - 1
21.         ListaCampi.AddItem FOGLIOXLS.Fields(CONTA).Name
22.     Next CONTA
23.     ListaCampi.ListIndex = 0
24.     PulsanteAggiungiSezione.Enabled = True
25.     PulsanteEliminaSezione.Enabled = False
26. End Sub
27.

```

Al caricamento del form vengono inseriti nella *ListBox* **ListaCampi** i nomi dei campi del foglio di Excel (riga 21) e la selezione del primo elemento (riga 23).

```

28. Private Sub PulsanteAggiungiSezione_Click()
29.     Dim TMPSEZIONE As TIPOSEZIONE
30.
31.     If FileTracciato.SelLength > 0 Then
32.         TMPSEZIONE.INIZIO = FileTracciato.SelStart
33.         TMPSEZIONE.FINE = TMPSEZIONE.INIZIO + FileTracciato.SelLength - 1
34.         TMPSEZIONE.CAMPO = ListaCampi.ListIndex
35.         TMPSEZIONE.RIGAFISSA = RigaFissaCheck.Value
36.         TMPSEZIONE.LUNGHEZZAFISSA = LunghezzaFissaCheck.Value
37.         TMPSEZIONE.ALLINEAMENTO = AllineaSinistraCheck.Value
38.         TMPSEZIONE.FILLER = FillerText.Text
39.         If ListaSezioni.ListIndex >= 0 Then
40.             SEZIONI.Add Sezione2Riga(TMPSEZIONE), , , ListaSezioni.ListIndex + 1
41.         Else
42.             SEZIONI.Add Sezione2Riga(TMPSEZIONE)
43.         End If
44.         FileTracciato.SelColor = RGB(255, 0, 0)
45.     End If
46.     MostraSezioni
47.     FileTracciato.SetFocus
48. End Sub
49.

```

Il click sul pulsante **PulsanteAggiungiSezione** comporta la creazione di una nuova sezione, i cui valori sono determinati dallo stato di alcuni elementi dell'interfaccia. Per creare una sezione è necessario selezionare del testo (almeno un carattere - riga 31) e premere questo pulsante. Verranno memorizzate le coordinate iniziali e finali della selezione (righe 32 e 33), il campo associato del foglio (riga 34), ed altri parametri definiti dai controlli posti nella parte centrale dell'interfaccia (righe 35-38).

Tale sezione verrà inserita subito dopo la sezione selezionata al momento nella *ListBox* **ListaSezioni** (riga 40) o all'inizio della *Collection* SEZIONI (riga 42).

Una volta creata la sezione il testo selezionato diverrà rosso (riga 44) al fine di poterlo distinguere dalle parti non selezionate.

Fin qui abbiamo inserito la nuova sezione nella *Collection*; la riga 46 provvederà a mostrare i dati della *Collection* nella *ListBox* **ListaSezioni** e poi riportare il focus sulla *RichTextBox* **FileTracciato** (riga 47).

[Segue parte 4 >>](#)



[Torna all'introduzione delle Richieste dei lettori](#)
