

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

http://www.vbsimple.net/activity/act_11_5.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, estragga 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 4](#)

Chiarito il funzionamento dell'utility di creazione dei files di definizione, possiamo completare la visione del form **MainForm**.

```
29. Private Sub GeneraOutput_Click()  
30.     Dim FILENR As Integer  
31.     Dim BUFFER As String  
32.     Dim SEZIONI As Collection  
33.     Dim TRACCIATO As String  
34.     Dim OGNISEZIONE As Integer  
35.     Dim PRIMORECORD As Boolean  
36.     Dim SEZIONEATTUALE As TIPOSEZIONE  
37.     Dim PUNTOLETTURA As Long  
38.
```

La routine legata all'evento Click sul pulsante **GeneraOutput** è la più complessa del form. Sarà necessario infatti, leggere il file del tracciato, il file di definizione e scrivere il file di output.

Per semplificare un po' le cose memorizzeremo tutte le sezioni in una *Collection* **SEZIONI** ed il tracciato in una variabile di nome **TRACCIATO**. Sarà necessaria anche una variabile di nome **PRIMORECORD** per indicare la lettura del primo record del foglio Excel, allo scopo di scrivere quelle righe fisse, che andranno scritte solo una volta.

```
39.     On Error GoTo ERRORE  
40.     If FileEsiste(DatiInputText.Text) = False Then Exit Sub  
41.     If FileEsiste(TracciatoText.Text) = False Then Exit Sub  
42.     If FileEsiste(DefinizioneText.Text) = False Then Exit Sub  
43.     If Len(Trim(NomeFoglioText.Text)) = 0 Then Exit Sub  
44.     If FileEsiste(OutputText.Text) = True Then Exit Sub  
45.
```

Prima di iniziare l'elaborazione ci assicuriamo che i file di input, il file del tracciato ed il file di definizione esistano (righe 40-42); inoltre controlleremo che sia stato inserito il nome del foglio Excel e che il file di output non esista (righe 43-44). Infatti la scrittura del file di output non avverrà se una di queste condizioni non si verifica.

```

46. Set SEZIONI = New Collection
47.
48. FILENR = FreeFile
49. Open DefinizioneText.Text For Input As FILENR
50. Input #FILENR, BUFFER
51. If UCase(Left(BUFFER, 19)) <> "FIBIA DATA STRIPPER" Then Exit Sub
52. While Not EOF(FILENR)
53.     Input #FILENR, BUFFER
54.     SEZIONI.Add BUFFER
55. Wend
56. Close FILENR
57. OrdinaSezioni SEZIONI
58.

```

Come prima operazione prepareremo la *Collection* **SEZIONI** istanziando la classe (riga 46), aprendo il file (righe 48-49), controllando che l'intestazione del file sia valida (riga 51) e leggendo una per una le sezioni dal file di definizione, memorizzandole nella *Collection* (righe 52-55). Al termine della lettura, le sezioni saranno ordinate (riga 57).

```

59. FILENR = FreeFile
60. Open TracciatoText.Text For Binary As FILENR
61. TRACCIATO = Space(LOF(FILENR))
62. Get #FILENR, , TRACCIATO
63. Close FILENR
64.

```

Fatto questo passeremo alla lettura del file di tracciato. Tale lettura avverrà in una istruzione, preparando prima il buffer **TRACCIATO** (riga 61) e poi leggendo l'intero file.

```

65. SEZIONEATTUALE.INIZIO = Len(TRACCIATO) + 10
66. SEZIONEATTUALE.FINE = Len(TRACCIATO) + 10
67. SEZIONEATTUALE.CAMPO = 0
68. SEZIONEATTUALE.RIGAFISSA = 0
69. SEZIONEATTUALE.LUNGHEZZAFISSA = 0
70. SEZIONEATTUALE.ALLINEAMENTO = 0
71. SEZIONEATTUALE.FILLER = "0"
72. SEZIONI.Add Sezione2Riga(SEZIONEATTUALE)
73.

```

Aggiungeremo anche una *SEZIONE* finta alla *Collection* delle sezioni per effettuare le corrette operazioni di posizione. Tale sezione infatti sarà 10 caratteri più ampia del file tracciato, avendo così la sicurezza che ogni altra sezione sarà minore di quella inserita per ultima.

```

74. FILENR = FreeFile
75. Open OutputText.Text For Binary As FILENR
76. If FILEXLS Is Nothing Then Set FILEXLS = OpenDatabase(DatiInputText.Text, 0, 0,
"Excel 5.0")
77. If FOGLIOXLS Is Nothing Then Set FOGLIOXLS = FILEXLS.OpenRecordset
(NomeFoglioText.Text & "$")
78. FOGLIOXLS.MoveFirst
79. PRIMORECORD = True

```

Prima di entrare nel ciclo di generazione del file di output prepareremo i file, effettuando l'apertura del file di output (righe 74-75), e del file e foglio di Excel (se essi non sono stati già aperti dall'utility di definizione), lo spostamento sulla prima riga e l'inizializzazione della variabile **PRIMORECORD** (righe 76-79).

```

80. While Not FOGLIOXLS.EOF
81.     PUNTOLETTURA = 0
82.     For OGNISEZIONE = 1 To SEZIONI.Count
83.         Riga2Sezione SEZIONI.Item(OGNISEZIONE), SEZIONEATTUALE

```

```

84.         BUFFER = ""
85.         If SEZIONEATTUALE.INIZIO > 0 Then BUFFER = Mid(TRACCIATO, PUNTOLETTURA + 1,
SEZIONEATTUALE.INIZIO - PUNTOLETTURA)
86.         Put #FILENR, , BUFFER

```

Per ogni record del foglio Excel sarà necessario leggere l'intero file tracciato e tutte le sezioni, e man mano, scrivere quei segmenti del file tracciato che non sono sezioni o meglio dire quelle parti del tracciato che precedono l'inizio di una sezione, a partire dal punto di lettura (righe 83-86).

```

87.         If (SEZIONEATTUALE.RIGAFISSA = 1) Imp (PRIMORECORD = True) Then
88.         BUFFER = Mid(TRACCIATO, SEZIONEATTUALE.INIZIO + 1, SEZIONEATTUALE.FINE -
SEZIONEATTUALE.INIZIO + 1)

```

Quella che vediamo alla riga 87 è un'operazione di implicazione logica tra due valori. Per molti programmatori potrebbe sembrare un'operazione nuova, non applicabile ad altri linguaggi direttamente. Per ulteriori informazioni sull'operazione di implicazione logica controllare la pagina dedicata alle [informazioni aggiuntive sull'algebra booleana](#).

Nel nostro caso verranno eseguite le operazioni all'interno del ciclo IF quando la sezione in analisi non tratta una riga fissa e non ci troviamo in presenza del primo record, ma verrà eseguita ogni volta che la sezione non è una riga fissa.

Così, leggeremo il testo dal file tracciato corrispondente alla sezione in analisi e lo memorizzeremo nella variabile BUFFER (riga 88).

```

89.         If SEZIONEATTUALE.LUNGHEZZAFISSA = 0 Then
90.         If SEZIONEATTUALE.CAMPO > 0 Then BUFFER = FOGLIOXLS.Fields
(SEZIONEATTUALE.CAMPO - 1).Value

```

Se la sezione in analisi non è a lunghezza fissa basterà sostituire le "XXX" della sezione (variabile BUFFER) con il contenuto del campo associato

```

91.         Else
92.         If SEZIONEATTUALE.ALLINEAMENTO = 1 Then
93.         If Len(BUFFER) > Len(FOGLIOXLS.Fields(SEZIONEATTUALE.CAMPO - 1).Value)
Then
94.         BUFFER = String(Len(BUFFER) - Len(FOGLIOXLS.Fields
(SEZIONEATTUALE.CAMPO - 1).Value), SEZIONEATTUALE.FILLER) & Left(FOGLIOXLS.Fields
(SEZIONEATTUALE.CAMPO - 1).Value & "", Len(BUFFER))

```

Se invece la sezione è a lunghezza fissa sarà necessario verificare se il dato estratto dal tracciato è maggiore del dato estratto dal foglio Excel. In questo caso sarà necessario controllare l'allineamento che i dati devono avere. Se l'allineamento è a sinistra, saranno riempiti gli spazi mancanti sulla destra con il carattere specificato nel campo **FILLER** della sezione.

```

95.         Else
96.         BUFFER = Left(FOGLIOXLS.Fields(SEZIONEATTUALE.CAMPO - 1).Value & "",
Len(BUFFER))
97.         End If

```

Se invece la lunghezza dei dati è minore o uguale alla lunghezza dei dati estratti dal foglio di Excel, non sarà necessaria alcuna operazione; basterà leggere tanti caratteri per quanto ampia è la sezione (riga 96).

```

98.           Else
99.           If Len(BUFFER) > Len(FOGLIOXLS.Fields(SEZIONEATTUALE.CAMPO - 1).Value)
Then
100.           BUFFER = Left(FOGLIOXLS.Fields(SEZIONEATTUALE.CAMPO - 1).Value & "",
Len(BUFFER)) & String(Len(BUFFER) - Len(FOGLIOXLS.Fields(SEZIONEATTUALE.CAMPO -
1).Value), SEZIONEATTUALE.FILLER)

```

Invece, se l'allineamento dei dati è a sinistra, dovremo controllare la lunghezza dei dati; se la lunghezza della sezione è maggiore della lunghezza dei dati, riempiamo lo spazio di differenza inserendo a destra il carattere specificato nel campo **FILLER** della sezione.

```

101.           Else
102.           BUFFER = Left(FOGLIOXLS.Fields(SEZIONEATTUALE.CAMPO - 1).Value & "",
Len(BUFFER))
103.           End If
104.           End If
105.           End If
106.           Put #FILENR, , BUFFER
107.           End If
108.           PUNTOLETTURA = SEZIONEATTUALE.FINE + 1
109.           Next OGNISEZIONE

```

Se invece la lunghezza non è maggiore della lunghezza dei dati basterà estrarre tanti caratteri dal foglio Excel per quanto è ampia la sezione.

Una volta preparati i dati che sostituiranno le "XXX" della sezione, potremo andare a scrivere i dati nel file di output (riga 106) ed incrementare la posizione di lettura del file tracciato, per permettere l'analisi delle altre sezioni.

```

110.           FOGLIOXLS.MoveNext
111.           PRIMORECORD = False
112.           Wend

```

Terminata la scansione di tutte le sezioni, si passerà alla riga successiva del foglio di Excel, ed imposteremo anche il valore della variabile **PRIMORECORD** a False. In tal modo, tutte le sezioni fisse non saranno riscritte, ma verranno evitate completamente.

Questa serie di cicli analizzerà tutte le righe del foglio di Excel e per ogni riga effettuerà le sostituzioni per tutte le sezioni.

```

113.           FOGLIOXLS.Close
114.           FILEXLS.Close
115.           Set FOGLIOXLS = Nothing
116.           Set FILEXLS = Nothing
117.           Close FILENR
118.           Set SEZIONI = Nothing
119.           MsgBox "Generazione file di output completata!", vbInformation + vbOKOnly,
"Genera output"
120.           Exit Sub
121.           ERRORE:
122.           On Error Resume Next
123.           MsgBox "Si è verificato un errore.", vbCritical + vbOKOnly, "Genera output"
124.           Set SEZIONI = Nothing
125.           If FreeFile > FILENR Then Close FILENR
126.           Set FOGLIOXLS = Nothing
127.           Set FILEXLS = Nothing
128.           End Sub

```

Finite le righe del foglio Excel potremo andare a chiudere il foglio, il database Excel, deallocare le variabili, chiudere il file di output, svuotare la Collection e completare

l'operazione di generazione del file di output con una finestrella che informa dell'avvenuta elaborazione (righe 113-120).

Ogni errore nella generazione del file di output porterà la visualizzazione di un messaggio di errore e la chiusura dei vari files.

Le operazioni viste non sono molto complesse, ma le istruzioni per effettuarle sono parecchio complicate. Si consiglia pertanto di effettuare un'operazione di traccia passo passo durante l'elaborazione del file di output.

La prova del progetto mostrerà il form principale. Prima di provare la generazione del file di output sarà necessario avere almeno un file di definizione per il tracciato scelto.

Pertanto, dopo aver inserito i dati del file di input, il nome del foglio Excel ed il percorso del file tracciato, possiamo premere il pulsante **Crea Definizione**.

Sarà richiamata l'utility per la definizione del file di definizione del tracciato, con estensione .FDF.

Una volta creato tale file, si potrà completare il processo di generazione del file di output.

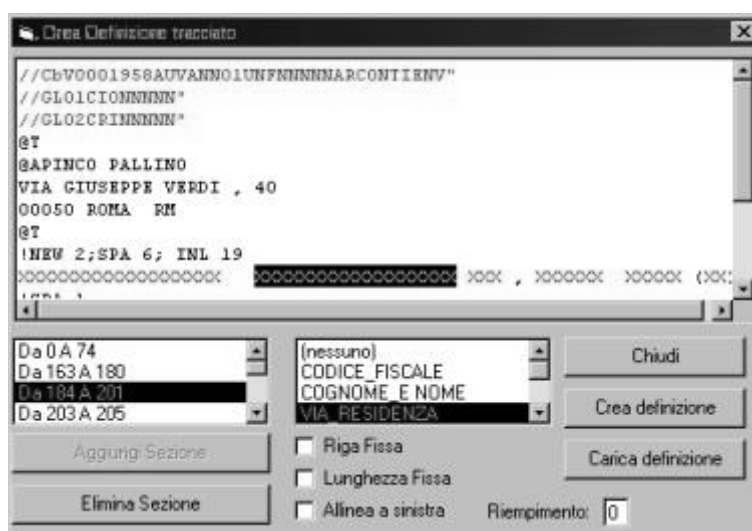


Figura 2

La finestra dell'utility di definizione del file tracciato apparirà come quella mostrata nella figura 2. Basterà selezionare dei segmenti del testo dalla finestra posta nella parte superiore, scegliere le impostazioni per la sezione dalla parte centrale del form, ed aggiungere ogni volta una sezione per il segmento selezionato.

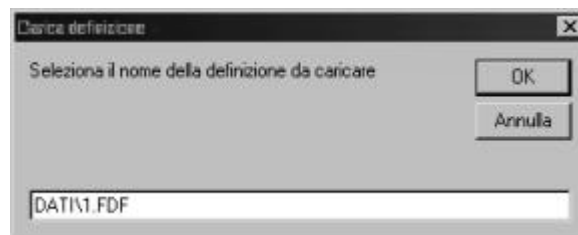
Alla fine della preparazione delle sezioni basterà premere il pulsante Crea Definizione per generare il file FDF scelto nella finestra che richiede il nome del file in cui salvare la definizione.



Ricordiamo che il salvataggio non effettua alcuna sovrascrittura dei files; pertanto è

necessario che il file con cui si tenta il salvataggio non esista.

Se vogliamo richiamare un file di definizione tracciato salvato in precedenza con quest'utility basterà specificare la posizione del file FDF da caricare.



Automaticamente saranno create le sezioni per tale file e sarà possibile aggiungere o eliminare sezioni a seconda delle necessità. Alla fine dell'operazione però, sarà necessario salvare il file con un altro nome, per i motivi accennati prima.

Terminate le operazioni riguardanti i files di definizione del tracciato, possiamo tornare al nostro form principale, chiudendo l'utility di definizione dei tracciati, oppure premendo in essa il pulsante **"Chiudi"**.



Basterà premere il pulsante **"Genera output"** per avviare l'elaborazione.

Se l'elaborazione procede senza errori, apparirà una finestra di informazioni che avverte che la generazione del file di output è completata.



La mancata apparizione di questa finestra indica che almeno un dato nel form principale è errato. Controllare pertanto i dati inseriti e riprovare a lanciare la generazione del file di output.

Il programma, giunto alla fine, è abbastanza complesso e richiede uno studio approfondito per carpirne a fondo il funzionamento.

Può essere uno spunto per effettuare operazioni sui files, in lettura e in scrittura, per effettuare un buon [parsing](#) (analisi di un file) e per effettuare una semplice connessione ad un database mediante l'utilizzo di un driver [ISAM](#).

[Fibia FBI](#)

27 Gennaio 2001



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