

# Visual Basic Intermedio

## Accesso a tutti i tipi di file, tranne i Database

Siamo arrivati al quarto articolo della serie dedicata al Visual Basic.

Nel primo abbiamo parlato di griglie ed affini (oggetti "visibili" che ospitano insiemi di dati), nel secondo di array e collezioni (elementi "non visibili" che ospitano anch'essi insiemi di dati) e nel terzo abbiamo parlato dello sviluppo di un'applicazione su più Form.

In questo quarto articolo parliamo dell'input/output di file di vario genere (file con dati testuali e file con dati grafici), nei due successivi parleremo dei possibili rapporti tra il Visual Basic ed i database, argomento che è, ovviamente, il più importante di tutti.

### Quarta parte

Ripetiamo, per chiarire ancora una volta il taglio degli articoli, quanto già detto in una puntata precedente. Si tratta di una serie di articoli dedicati agli utilizzatori del Visual Basic, articoli di "taglio intermedio", pensati per una precisa categoria di utilizzatori, non quelli alle prime armi né quelli già esperti, ma quelli che stanno in mezzo, che hanno già le conoscenze di base e che vogliono approfondire solo alcuni temi particolari.

Gli esperti invece già dovrebbero conoscere gli argomenti trattati e quindi dovrebbero anche essere in grado di svolgere facilmente gli esercizi che corredano l'articolo... ma non è detto.

Gli articoli sono prevalentemente pratici, nel senso che propongono una serie di esercizi facilmente rieseguibili da ciascuno di voi. Anche in questo terzo

articolo ogni esercizio corrisponde ad una figura che mostra sia il listato sia, in primo piano, il Form, che costituisce l'aspetto esteriore dell'applicazione.

### Ecco il nostro piano per questo e per i successivi due articoli

Questo articolo parla di come da Visual Basic si possano leggere file, file testuali di tutte le categorie, "poveri" e "ricchi", file grafici di tutte e due le categorie esistenti (BitMap e vettoriali) e delle varie sottocategorie, file di altro tipo, ad esempio quelli multimediali.

Saremo assolutamente sistematici, proponendovi una dozzina di esercizi in

cui leggeremo i vari file e li utilizzeremo in qualche maniera, infilandoli ad esempio in qualche controllo in grado di ospitarli. Diamo per scontato che questi controlli siano da voi ben conosciuti, la nostra attenzione essendo rivolta soprattutto a come leggere i file.

### Prepariamo i file di prova

Prima di eseguire la sperimentazione, è opportuno preparare una serie di file di prova, il cui contenuto e le cui caratteristiche ci siano ben note, in modo da verificare senza ambiguità (errore nel programma o errore nei dati?) il buon funzionamento della nostra applicazione.

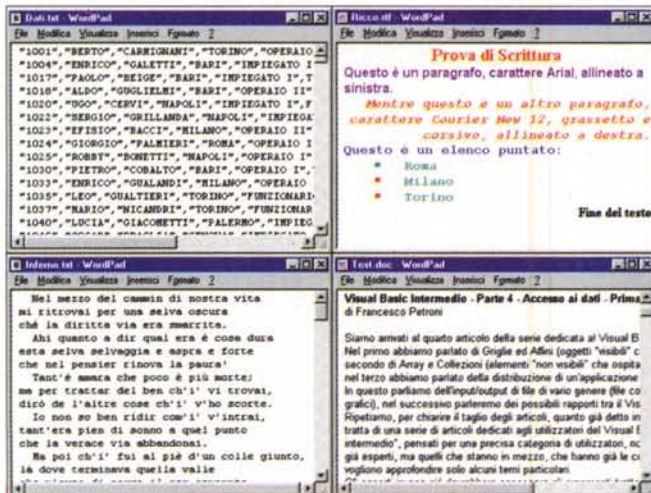


Figura 1 - Visual Basic 5.0 - Alcuni dei file che leggeremo.

In questo articolo vedremo come, con Visual Basic, si possano leggere file di tutti i tipi, testuali semplici, testuali ricchi, grafici bitmap e vettoriali, multimediali, ecc. A partire dal prossimo numero ci dedicheremo alle metodologie, ricchissime, di accesso ai dati con Visual Basic. Nella figura vediamo quattro dei file che proveremo a leggere. Un testuale puro, di tipo "comma delimited", un altro testuale puro da leggere riga per riga, in quanto non di tipo "delimitato". Un file RTF, che è un formato testuale pompato con codici di formattazione abbastanza chiari ed interpretabili (Rich Text Format). In basso a destra ecco a voi un file in formato Word e quindi DOC. Di questo abbiamo anche una versione TXT puro.

Prepariamo quindi una serie di file testuali e grafici, che vediamo, in parte, nelle prime due figure (figura 1 e 2). Ne elenchiamo le estensioni e ne descriviamo le caratteristiche:

DATI .TXT	file sequenziale di tipo "comma delimited" con tenente dati
INFERNO .TXT	file sequenziale, non delimited, da leggere per righe
TEST .TXT	file testuale, non delimited, da leggere tutto in sieme
TEST .DOC	versione, in formato Word, del precedente
RICCO .RTF	file molto formattato e salvato nel formato RTF
TEST .HTM	file in formato HTML, con link e codice VBS
DATI .XLS	file Excel con una tabellina di dati

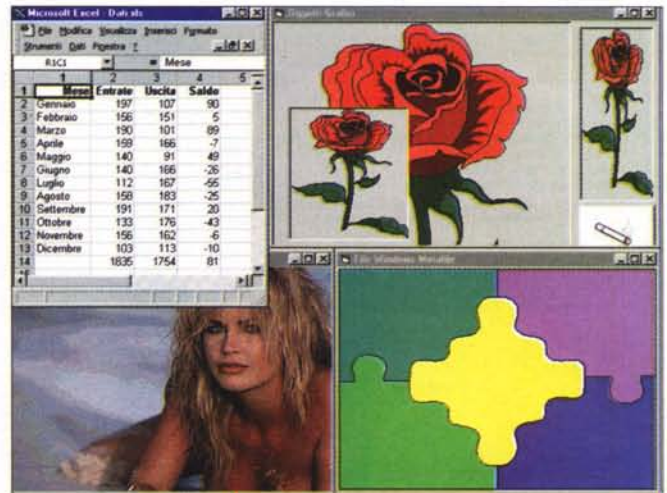


Figura 2 - Visual Basic 5.0 - Alcuni dei file che leggeremo.

Per quanto riguarda la lettura dei file grafici, Visual Basic è ben attrezzato riguardo i file Bitmap (legge tutti i tipi) mentre, per quanto riguarda i file vettoriali, si limita alla lettura dei WMF. Gli oggetti in grado di ospitare file grafici sono tanti, sia i Form che le PictureBox, che le ImageBox. Inoltre anche pulsanti e toolbar possono assumere un aspetto grafico e pure la FlexGrid può mostrare figure. Esistono centinaia di controlli OCX di terze parti che permettono anche una certa manipolazione dei file grafici. Ad esempio molti di voi dispongono del programma della Wang Imaging per Windows 95, che, essendo realizzato con il Visual Basic, si porta dietro ben quattro controlli aggiuntivi: provate a sfruttarne i servizi. Nella figura abbiamo anche inserito un foglio Excel, che utilizzeremo per ulteriori esperimenti.

ROSA .WMF	file ClipArt di Office in formato Windows Metafile
PUZZLE .WMF	come sopra
* .WMF	vari file Windows Metafile
* .BMP	vari file BitMap, come gli sfondi di Windows
* .JPG	il paginone centrale di Playboy disponibile ogni mese su Internet
* .ICO	file con le Icone, disponibili con Visual Basic
* .MDI	file con spezzoni audio Midi
* .WAV	file con spezzoni audio WAV
* .AVI	file con spezzoni video AVI
* .ASF	file con spezzoni video stream.

Per ognuno di questi tipi di file, Visual Basic propone metodi di lettura e oggetti in grado di ospitarli.

## Letture di file dati e di file testuali

Per quanto riguarda i file testuali, proponiamo tre esercizi con tre varianti di modalità di lettura, ognuna delle quali è adatta per uno specifico tipo di file.

Il file sequenziale DATI.TXT, che leggiamo nel primo esercizio (figura 3), è del tipo "comma delimited": contiene

dati, ogni riga è un record, ogni record contiene lo stesso numero di campi separati da virgole. Se il campo è di tipo alfanumerico il dato è racchiuso tra virgolette, altrimenti niente virgolette.

La classica routine di lettura è la seguente:

```
Open "nomefile" For Input As #1
Do While Not Eof(1)
  Input #1, X0,X1.. Xn
  ... omissis
```

Loop

Close #1

#1 indica il primo "canale" nel quale si legge (il fatto è che potrebbero essere aperti più canali, ad esempio due in lettura ed uno in scrittura). Eof(1) è una funzione che risponde Vero quando i dati nel primo canale sono finiti. Do While ... Loop è la routine di scorri-

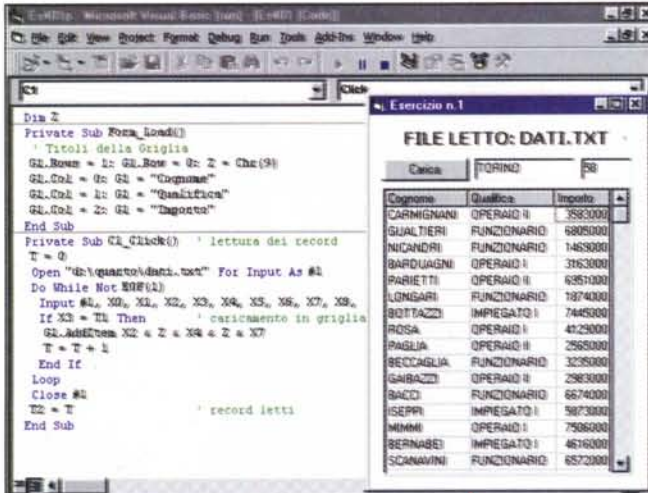


Figura 3 - Visual Basic 5.0 - Lettura di un file sequenziale.

La lettura più orientata ai dati è sicuramente quella dei file testuali "comma delimited", in quanto si tratta di un formato riproducibile con qualsiasi sistema e con qualsiasi prodotto. Il file contiene solo i dati e quindi l'organizzazione del file (nome dei campi, tipo, lunghezza, ecc., in termini tecnici: il tracciato record) deve essere nota a chi cerca di leggerlo. Qui vediamo la classica routine di scorrimento e di lettura del file. Letto il singolo record, che contiene 10 campi, viene eseguito un test sul campo che contiene la sede, e viene caricata una griglia con solo 3 dei 10 campi, che vanno comunque letti tutti.

mento.

La lettura vera e propria del record è eseguita dall'istruzione Input #1, ... che pretende tante variabili quanti so-

no i campi del record. L'istruzione Input provvede anche a spostare il puntatore sul record successivo. ... omissis indica la porzione di codice che si occupa di utilizzare i dati appena letti e piazzati nelle variabili.

Abbiamo detto che è necessario che le variabili in lettura siano tante quanti sono i campi letti, anche se di questi se ne utilizzano di meno. Se si leggono meno variabili, l'istruzione Input mischia le righe, i dati si scomporgono ed alla fine appare l'errore "Input Paste

End", che significa che l'ultimo record non è stato letto completamente.

Nel nostro primo esercizio (figura 3) i dati letti vengono parzialmente (10 campi letti, 3 campi usati) scaricati in una griglia. Vengono anche filtrati sulla base di una città da indicare nella prima TextBox.

Il secondo esercizio (figura 4) mostra una variante dell'istruzione Input, e cioè la Line Input, che va usata quando il file non prevede delimitazioni. Nel nostro caso, lettura della Divina Commedia, ci sono versi con virgole, una, due, ecc., e versi senza virgole, per cui la riga va letta tutta intera. Le righe, pardon i versi, li scarichiamo in una ListBox, che poi leghiamo ad un timer per farla scorrere automaticamente.

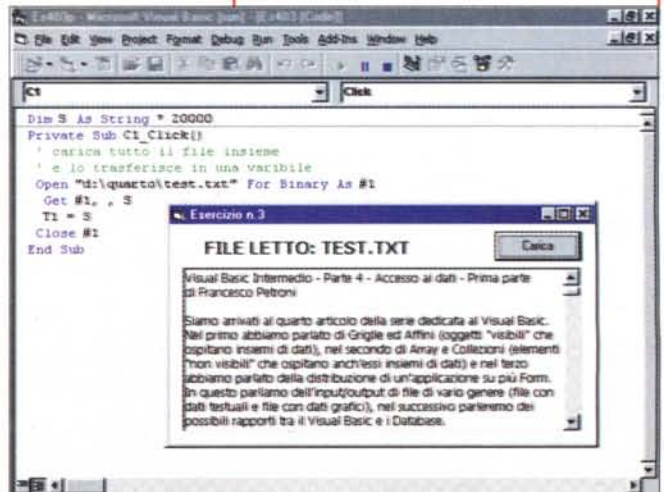
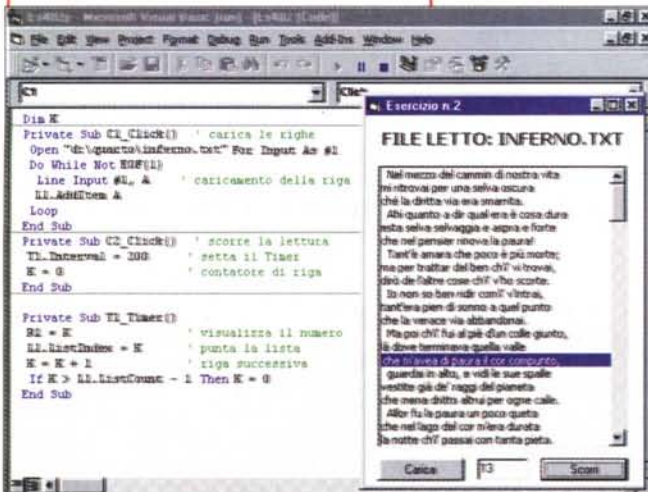
Il terzo esercizio (figura 5) mostra come sia possibile leggere un file testuale qualsiasi, tutto insieme, e piazzarlo, tutto insieme, in una variabile. C'è un limite teorico alla dimensione della va-

Figura 4 - Visual Basic 5.0 - Lettura riga per riga di un file sequenziale.

Dante Alighieri, quando scrisse la Divina Commedia, era indeciso se utilizzare un file sequenziale "comma delimited" o uno "fixed length". Dal nostro punto di vista di utilizzatori di file, il sistema più produttivo per leggere un file fatto di versi (e quindi di righe) è quello di utilizzare l'istruzione Line Input che legge tutta la riga, virgole comprese. Nell'esercizio, oltre a leggere i versi ed a piazzarli in una ListBox, abbiamo anche attivato un timer che fa avanzare via via il verso puntato, in modo che chi legga i versi non debba avanzare "a mano" con la scrollbar della ListBox.

Figura 5 - Visual Basic 5.0 - Lettura, in blocco, di un file testuale.

Estremizziamo il discorso, iniziato con l'esercizio precedente, leggendo un file testuale normale, ottenuto salvando il file DOC, che vedete in figura 1, in formato TXT. In questo caso ovviamente non si tratta né di un file "comma delimited" né di un file leggibile riga per riga: non esistono campi, non esistono righe, al massimo esistono "paragrafi" di lunghezza qualsiasi. Leggendolo come Binary lo si legge tutto in blocco e lo si può piazzare, tutto insieme e tutto intero, in una variabile da dichiarare come stringa lunga (es. Dim S As String \* 20000). La variabile può essere ospitata in una TextBox da impostare come Multiline.



riabile di 64 kbyte, ma il limite pratico è ben inferiore. Il listato è brevissimo:

```
Open "nomefile" For Binary As #1
Get #1, , S
Close #1
```

Le due virgole dell'istruzione Get servono per un eventuale numero che indica da che carattere iniziare a leggere. Nessun numero, si inizia dal primo carattere.

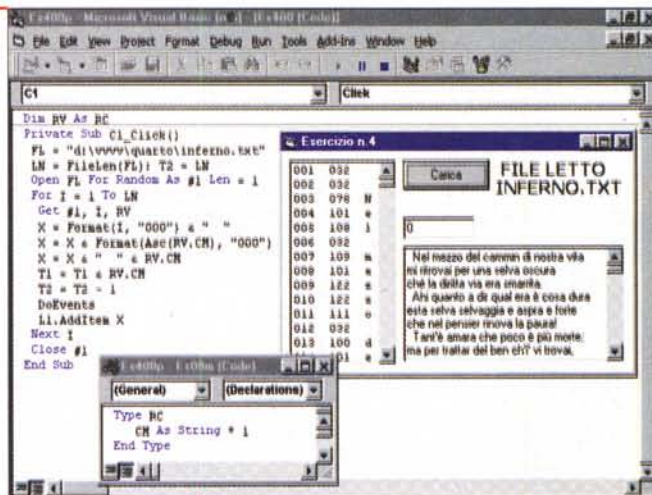
## Ulteriori modalità di lettura

Un altro tipo di file molto diffuso è quello che si chiama "fixed length" e che prevede che i campi siano piazzati in posizioni fisse e che lo spazio tra i campi sia riempito con degli spazi. Es.:

```
ROSSI.....UGO.....333333
BIANCHI...MARCELLO.....222
```

Figura 6 - Visual Basic 5.0 - E se dovessimo leggere un file "byte per byte"?

Non trattiamo la più sofisticata tra le modalità "tradizionali" di lettura dei file dati: quella che consente un accesso random anziché un accesso sequenziale (in pratica si raggiunge il record desiderato senza dover leggere tutti quelli precedenti). La modalità di lettura random pretende ovviamente la dichiarazione della struttura del record e conseguentemente la dichiarazione della sua lunghezza. Ovviamente ancora niente a che vedere con le tecnologie di accesso ai database, possibili con VB, che sono "letteralmente" impressionanti e delle quali parleremo a partire dal prossimo articolo.



po random, che permette di leggere e scrivere un record di un file, conoscendone la posizione numerica, senza dover leggere i record precedenti o successivi. Questo sistema necessita della

Figura 7 - Visual Basic 5.0 - Lettura di un file "ricco".

Prima della nascita del formato HTML esisteva un altro formato codificato, in cui, in altre parole, c'è del testo e ci sono dei codici di formattazione, il formato RTF. Si tratta di un formato non legato a questo o a quel prodotto, e quindi riconosciuto in scrittura ed in lettura da tutti i prodotti più importanti. Lo produce anche l'accessorio WordPad di Windows 95/98, che tutti possono usare. Qui vediamo sia come con Visual Basic si possa leggere un file RTF (esiste un apposito controllo OCX: Rich Text Box) e come lo stesso, letto invece come binario e posto in una TextBox, mostri i suoi codici. Si tratta di codici più complessi di quelli che ritroviamo in HTML.

```
NERI.....MARIO.....1111
```

Si legge tutta insieme l'intera riga e la si piazza in una variabile che poi va scomposta con una funzione Mid:

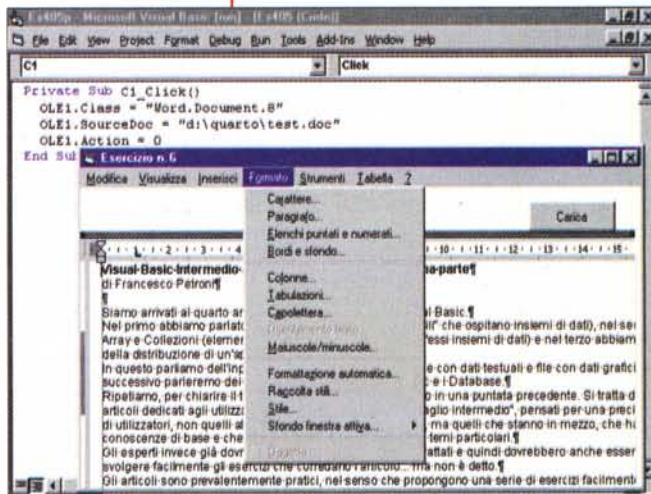
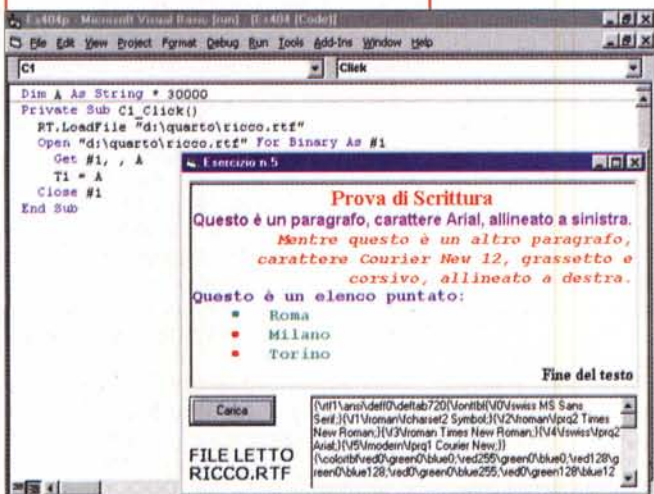
```
... omissis
Input #1, X
... omissis
```

```
Cognome = Left(X,10)
Nome = Mid(X,11,8)
Importo = Mid(X,19,10)
```

Non parliamo invece del sistema di scrittura e di lettura più sofisticato, il-

Figura 8 - Visual Basic 5.0 - Lettura di un file Word, messo in un OLE Container.

Non essendo possibile la lettura diretta di un file in formato Word, l'unico modo per accedere ad un tale tipo di file è quello di sfruttare OLE, o riempiendo un OLE Container con il file stesso o sfruttando OLE Automation per aprire Word e manipolare direttamente il contenuto del documento. Nel primo caso occorre solo impostare due o tre proprietà dell'oggetto contenitore (come si può vedere dal listato). OLE Automation invece è leggermente più complicato, lo utilizziamo nell'esercizio seguente per leggere "da dentro" un file Excel.



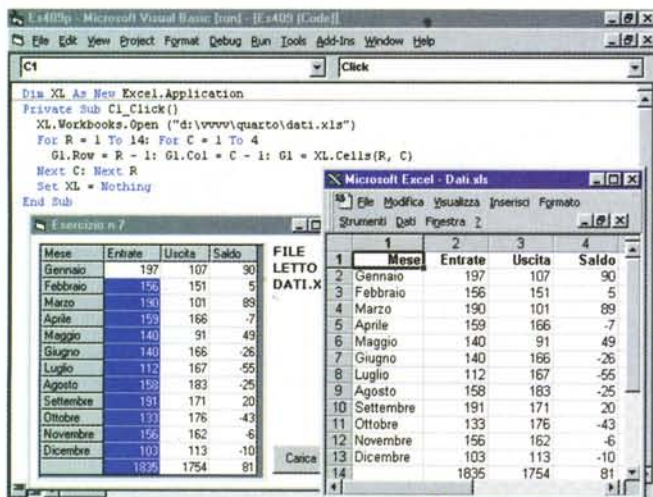


Figura 9 - Visual Basic 5.0 - Lettura, con OLE Automation, delle celle di un file Excel. Qui leggiamo, grazie alla potentissima tecnologia OLE Automation, il contenuto delle celle di un foglio Excel per riportarle nelle celle di una FlexGrid di Visual Basic. L'applicazione Visual Basic richiama, come componente ActiveX a tutti gli effetti, le librerie di Excel, e quindi il listato può contenere istruzioni VBA, che attivano l'oggetto e che poi operano all'interno dell'ambiente Excel.

dichiarazione della struttura e della lunghezza del record (in modo che la routine di lettura possa "calcolare" esattamente la posizione del record da leggere). Della struttura vanno dichiarati i campi e la loro tipologia, ad esempio i campi numerici possono essere di tipo Byte, Integer, Long, Single e Double, proprio come le variabili numeriche di Visual Basic.

Non esiste più motivo per usare questi tipi di file, per il semplice fatto che è molto più facile accedere ai database e che le funzionalità che lavorano con i dati del database sono molto più sofisticate in termini di possibilità di manipolazione ed efficienti in termini di prestazioni.

Ciò nonostante, vi proponiamo un esercizio che serve a leggere un file "byte per byte" e che in tal modo può leggere qualsiasi file. Lo vediamo in figura 6.

Va creato un file Bas, in cui viene dichiarata la struttura del file in lettura (un solo campo di un solo byte), poi nel programma di lettura viene dichiarata la lunghezza del record (un solo carattere)

e vengono letti i caratteri uno per uno. Questi vengono intabellati in una lista, con numero progressivo del carattere letto, codice ASCII e valore, e poi scaricati in una TextBox.

Nell'esempio mostrato abbiamo letto un file testuale, per cui appaiono prevalentemente caratteri noti. Altra musica se si leggono file molto codificati e file binari.

## Un tipo un po' speciale

Un tipo di file un po' passato di moda, per il fatto che si è ormai imposto l'HTML, è il Rich Text Format, che è un formato testuale "in chiaro", in cui il testo è formattato e le specifiche di formattazione corrispondono a codici ben identificabili. Un file RTF si può scrivere con un qualsiasi word processor moderno (basta il WordPad di Windows 95/98) e si può leggere (ed al limite anche scrivere) con il nostro Visual Basic 5.0.

Nel quinto esercizio (lo vediamo in figura 7) vi proponiamo due modi di leggere il file RTF, nel controllo Rich Text Box, che è in grado di interpretare i codici, e nel normale TextBox, che, non essendo in grado di interpretare i codici, li mostra.

Per chiarire meglio le differenze tra i due controlli "ospitanti", diciamo che la TextBox ha un suo proprio ed unico carattere di formattazione, mentre la Rich Text Box è in grado di mostrare al suo interno testi scritti con vari caratteri e le

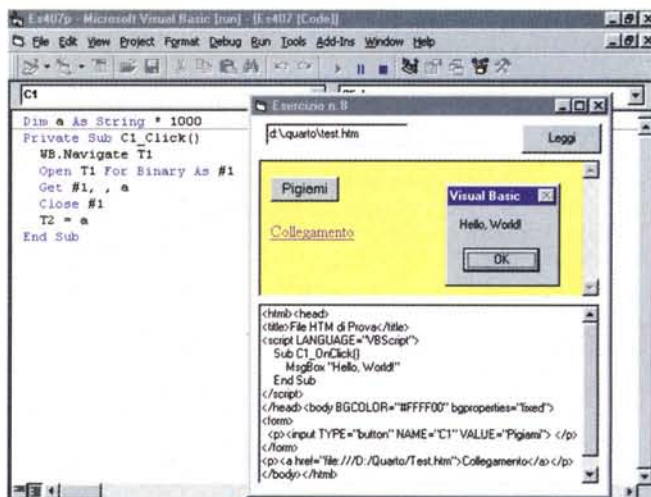


Figura 10 - Visual Basic 5.0 - Lettura di un file HTML.

Visual Basic 5.0 dispone di un controllo in grado di ospitare file di tipo HTML: Microsoft Internet Control e di riconoscere tutte le caratteristiche interne di tale formato, link compresi. Questo controllo permette di realizzare un vero e proprio browser personalizzato, completo di funzionalità, e magari controllato e limitato nelle possibilità di navigazione ai soli siti consentiti. Nell'esempio, vediamo come questo sia anche in grado di interpretare codice Visual Basic Script incorporato nel codice HTML. In basso, nella solita TextBox, vediamo il codice HTML completo in cui è inserita la routine VBS.

varie formattazioni.

## Materiale Office, documenti DOC o cartelle di lavoro Excel

Se, nella nostra applicazione Visual Basic, vogliamo visualizzare documenti realizzati con prodotti Office possiamo percorrere, in generale, due alternative. La prima è l'utilizzo dell'OLE Container, che può ospitare un documento Office, o qualsiasi file realizzato con un prodotto OLE Server, la seconda è l'utilizzo di OLE Automation, ovvero la programmazione, partendo da Visual Basic, del documento esterno.

Per quanto riguarda l'OLE Container (l'esercizio è il numero 6 e la figura la numero 8), questo può essere facilmente riempito da programma, impostando le proprietà Class e SourceDoc, ed



Figura 11 - Visual Basic 5.0 - Finestra File Apri e lettura di un file grafico. La finestra File Apri è un servizio di Windows facilmente utilizzabile, grazie al controllo Common Dialog Box che ne rende immediato (due o tre istruzioni) lo sfruttamento. Qui lo usiamo, dopo averlo opportunamente personalizzato, per permettere all'utente dell'applicazione di scegliere il file di tipo grafico da usare come sfondo (proprietà Picture) del Form. Si tratta del primo e più semplice esempio di utilizzo di un file grafico con VB.

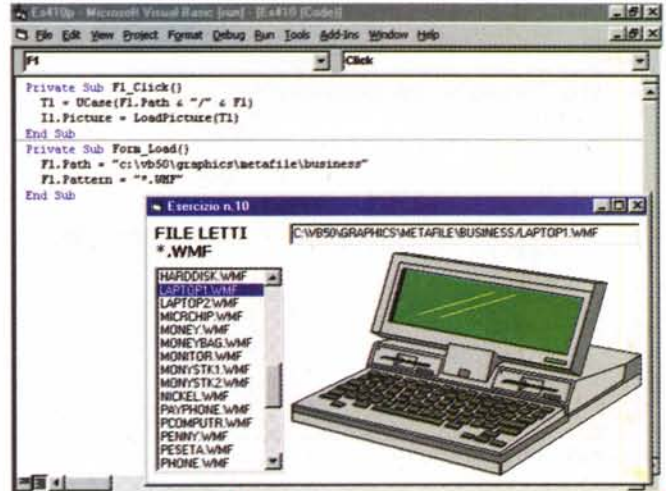


Figura 12 - Visual Basic 5.0 - Lettura di un file Windows Metafile. Visual Basic è un po' scarso per quanto riguarda la lettura dei file vettoriali, anche perché non esistono molti formati di tale categoria. Il più noto e diffuso è il Windows Metafile, il formato WMF (quello in alto a destra ed in basso a destra in figura 2). Poiché esistono molte librerie ClipArt in questo formato (ne hanno sia il Visual Basic stesso, che MS Office) non è difficile con il nostro Visual Basic realizzare applicazioni che permettano di consultarle. E' quello che abbiamo fatto. Il file in basso a destra in figura 2 lo abbiamo realizzato con PowerPoint, per evidenziare il fatto che nel formato grafico WMF vengono memorizzati anche gli effetti sfumatura.

eseguendo il metodo Action. Evidentemente l'oggetto DOC (nel caso, il nostro, in cui leggiamo un file Word) può essere editato "in-place".

Più "da programmatore" è la tecnologia OLE Automation (l'esercizio è il numero 7 e la figura la numero 9) grazie alla quale viene aperto un file XLS (nel caso, il nostro, in cui leggiamo un file Excel) e ne vengono maneggiati gli oggetti. Nel nostro esercizio, che non commentiamo, non facciamo altro che leggere il contenuto delle varie celle Excel per riportarle in una griglia VB.

## A proposito di HTML

Prima, parlando di RTF, abbiamo detto di come questo formato sia caduto in disgrazia a causa del successo dell'HTML. Ora vediamo come, dalla nostra applicazione Visual Basic, riusciamo a leggere proprio un file HTML (l'esercizio è il numero 8 e la figura la numero 10). Usiamo un Microsoft Internet Control, che fa diventare la nostra applica-

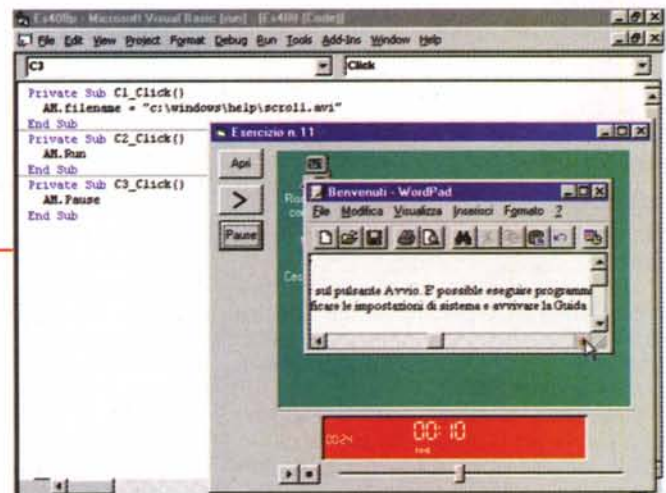
zione un browser Internet. La navigazione avviene semplicemente impostando la proprietà Navigation del nostro oggetto. E' da notare come il componente sia un browser compatibile con Explorer 3.01. Riconosce, e ci mancherebbe altro, i link, e riconosce anche il VBS, il Visual Basic Script. Lo vediamo nella figura. In particolare vediamo il pulsante sulla pagina WEB e la MessageBox provocata dal click sul pulsante. Nella TextBox in basso mostriamo anche il codice HTML completo.

## File grafici, benvenuti a bordo

La seconda parte dell'articolo la dedi-

Figura 13 - Visual Basic 5.0 - Non dimentichiamoci del multimediale.

Stesso discorso per i file MM, audio e video: Visual Basic è in grado di leggere e di utilizzare file MM. Esistono una serie di controlli che permettono non solo di caricare file MM, ma anche di "pilotarli" da programma. In figura vediamo il nuovo ActiveMovieControl della Microsoft, scaricabile dal sito. Uno dei miei favoriti è il controllo MCIWnd che permette, tanto per dire, anche di controllare la velocità di visualizzazione dello spezzone video o il volume della sezione audio.



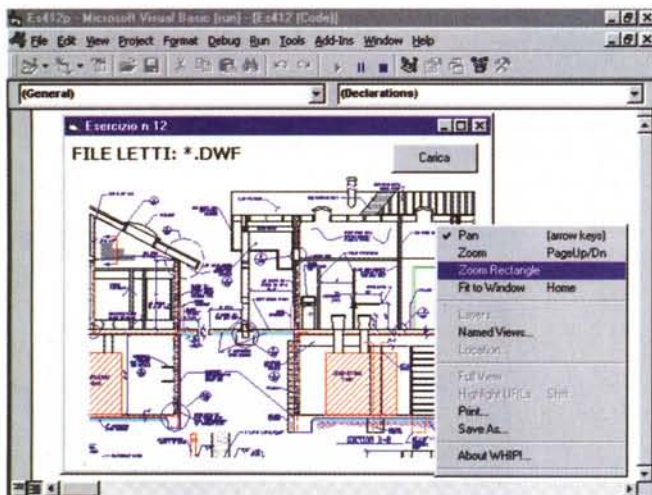


Figura 14 - Visual Basic 5.0 - Controlli aggiuntivi per leggere i file (file AutoCAD con il controllo Whip!).

A seguito dello "sconfinamento" della tecnologia COM su Internet, molte case si sono date da fare per realizzare componenti OCX bivalenti, pensati per la distribuzione su Internet, ma utilizzabili anche "in privato" come normali controlli OCX in applicazioni Visual Basic. Qui vediamo all'opera il famoso Whip! dell'Autodesk, che mostra file AutoCAD salvati in uno speciale formato leggero DWF. Il controllo dispone di un QuickMenu che permette di attivare una serie di funzionalità per la visualizzazione e la stampa del disegno.

chiamo ai file grafici, usabili in varie maniere e con vari oggetti, in Visual Basic. File grafici, quindi immagini, sia di tipo Bitmap che di tipo vettoriale, possono essere usati in molti oggetti. Ad esempio come sfondo di un Form, all'interno di controlli Picture o Image, all'interno di celle di una FlexGrid, su pulsanti normali o all'interno di una toolbar.

Negli esempi che vanno dal 9 e 10, e nelle figure 11 e 12, e anche nella 2, vediamo qualche esempio di utilizzo di file grafico.

Nel primo usiamo un controllo Common Dialog Box, per attivare la finestra di Windows File Apri nella quale scegliere un file che faccia da sfondo alla nostra finestra.

Nel secondo usiamo una File ListBox, che punta ad una cartella piena di file WMF (ad esempio i ClipArt di Office) in modo che sia facile scegliere e visualizzare i vari disegni.

In alto a destra nella figura 2 vediamo la stessa immagine, una rosa WMF, in tre dimensioni. Vogliamo semplicemen-

te sottolineare il fatto che un file grafico (di qualsiasi tipo) ha una sua dimensione di base, e che non è detto che lo spazio che gli lasciamo nella nostra applicazione VB sia adatto a quella dimensione.

## Varie ed eventuali

Mancano all'appello i file multimediali, AVI, ASF, per quanto riguarda quelli video, e MIDI e WAV, per quanto riguarda quelli audio. Anche in questo caso Visual Basic non si tira indietro. Mette a disposizione alcuni controlli con i quali si attivano e si controllano i file dei tipi citati. In figura 13 vediamo il componente ActiveMovie, recentissimo, che può essere facilmente usato, e due o tre proprietà, per pilotare lo spezzone.

Dobbiamo, ancora per completezza di informazione, parlare delle centinaia di controlli aggiuntivi, disponibili sotto forma di ActiveX (DLL o OCX) per Visual Basic, che permettono di leggere questo o quel file. In figura 14 vediamo l'OCX preparato dall'AutoDesk per permettere di visualizzare attraverso Internet i disegni fatti con AutoCAD. Si chiama Whip! ed in quanto ActiveX standard può essere usato, salvo i normali problemi di "autenticazione", anche in applicazioni VB.

Ultimissimo esempio è quello che riguarda l'utilizzo dei file di tipi ICO, quelli che contengono le icone. Possono essere visualizzati su una FlexGrid (come fare lo vediamo dal listato dell'ultimo esercizio, il 13 in figura 15) oppure

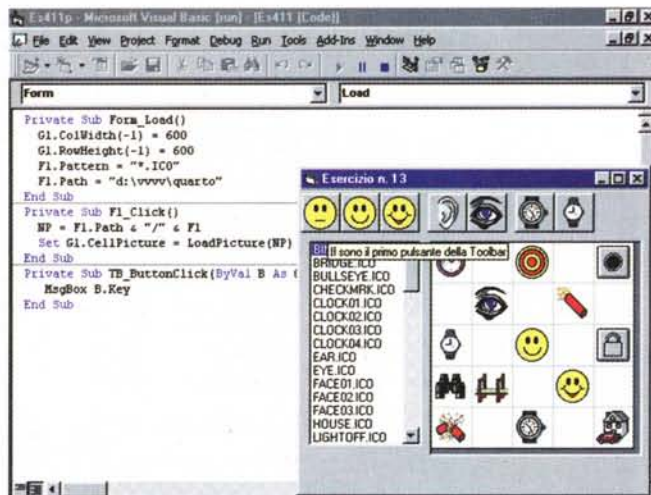


Figura 15 - Visual Basic 5.0 - Icone e bottoni a gogo.

In molte applicazioni realizzate con Visual Basic i vari file, testuali o grafici, vanno caricati per "motivi di servizio". A esempio le icone (salvate in un file ICO) possono essere usate per i Form, per i pulsanti delle toolbar, per i pulsanti "normali". Nel nostro esercizio le leggiamo sia per visualizzarle in una FlexGrid sia per caricarle in una ImageList, che poi le passa alla toolbar.

usati su una Toolbox, grazie all'intermediazione di un controllo di tipo ImageList.

Come noto l'oggetto ImageList serve a raccogliere le varie immagini, che vengono identificate con un numero progressivo, poi la Toolbox fa riferimento a quel particolare numero. Questa intermediazione, che potrebbe sembrare un passaggio inutile, in realtà rende il sistema più dinamico, rendendo possibile la modifica dinamica del contenuto della Toolbox.

## Prossimamente su queste pagine

Esaurita, pressoché completamente, la problematica relativa all'accesso ai file di tutti i generi, rimandiamo a un prossimo numero (ma ci vorranno un paio di puntate) la trattazione della problematica relativa all'uso dei database con Visual Basic.

WEBSHOW!'98. PER SCOPRIRE IL MONDO CHE CAMBIA.

# CHIOCCIOLA

Webshow! La Fiera dedicata all'Information & Communication Technology.

**FIERA DI ROMA 4/8 DICEMBRE 1998**

E' un appuntamento per tutti. Per chi sa già tutto e chi no. Un'occasione unica per scoprire tutte le nuove tecnologie informatiche, i computer della nuova generazione, viaggiare in Internet e interagire direttamente con tutte le novità del mondo che cambia. E' il mondo di Webshow! '98.

Inizia lo show! (Fiera di Roma, 4/8 dicembre dalle 10.00 alle 19.00)

[www.webshow.fieradiroma.it](http://www.webshow.fieradiroma.it)  
webshow@fieradiroma.it - Tel. 065178238



**webshow!**

COMUNICARE E INTERAGIRE DENTRO E FUORI LA RETE.



IN COLLABORAZIONE CON:



CoFax

BNL Multiservizi

UNIDATA

business solutions

Luiss

ORGANIZZAZIONE

GRUPPO EDITORIALE JACKSON



Radio Dimensione Suono

RADIO UFFICIALE