



## Microsoft Visual Basic 3.0

di Francesco Petroni

**I**l Visual Basic sta battendo tutti i primati di velocità. È uscito da pochi mesi (la prova della versione 1.0 è stata pubblicata nel numero 111, ottobre 1991, di MC, ed è stata seguita da un piccolo corso destinato più al vasto pubblico degli utenti finali «curiosi di Windows» che a quello degli specialisti in programmazione, dal numero 125 al numero 128).

Poco dopo ne è uscita la versione professionale. Dopo pochi mesi ecco la versione 2.0, nel modello «normale» e in quello «professionale». Neanche un attimo per rifiatore ed ecco la versione 3.0, subito anche questa nei due modelli.

Questo per tacere della parallela linea Visual Basic per DOS e del fratello maggiore Visual C++, anche questo linguaggio di visual programming, ma sta-

volta destinato ai soli «professional».

Molti sono stati e molti ancora saranno i motivi di interesse per il Visual Basic. Ne citiamo tre.

Innanzitutto l'interesse per il linguaggio in sé, che pur essendo lontanissimo erede del Basic, di Gatesiana memoria, è l'esempio oggi più accessibile (anche ad un utente mediamente esperto) di linguaggio Object Oriented.

Il secondo motivo è che Visual Basic è totalmente integrato in Windows. Con esso sono, in maniera molto facilitata, resi disponibili, o meglio resi programmabili, tutti gli oggetti Windows, a cui un utente Windows è abituato, e che quindi possono entrare facilmente a far parte della applicazione che si sta sviluppando.

Il terzo motivo di interesse è costituito dal fatto che la Microsoft ha chia-

ramente individuato nel Visual Basic il linguaggio «unico» di tutti i suoi prodotti futuri. In altre parole il «nucleo» del Visual Basic sarà l'unico linguaggio di programmazione di tutti i suoi applicativi per gli utenti.

Questo «piano» ambizioso sarà attuato nei prossimi anni.

Per ora abbiamo Access Basic, il linguaggio di Microsoft Access, pressoché identico al Visual Basic.

Word Basic, il linguaggio macro di Word 2.0, è invece ancora molto differente dal VB. L'altro prodotto a larga diffusione della Microsoft, per utente e programmabile, parliamo dell'Excel, invece dispone ancora di un linguaggio macro orientato allo spreadsheet. Le analogie con Visual Basic sono in questo caso minime.

Concludendo chiunque fosse inte-

ressato a Windows non solo come utente finale di uno o più prodotti Windows, ma come utente evoluto desideroso di sviluppare qualche cosa da solo, oppure come «professional», persona cioè che svolge una attività produttiva, per sé o per altri, usando il Personal Computer, deve conoscere il Visual Basic.

### Programming System for Windows

La frase che segue il termine Visual Basic è «programming system for Windows». Interessante è la presenza del termine «system» anziché del termine «language». Un «sistema» è più di un «linguaggio».

Nel sistema VB è compreso un linguaggio, derivato dal vecchio Basic, e sono comprese... altre cose.

Le altre cose sono i cosiddetti control, che potremo definire elementi prefabbricati, in cui non c'è più nulla da programmare. L'unico lavoro che l'utente deve svolgere è quello di adattare l'elemento prefabbricato alla propria applicazione.

Nel sistema sono compresi anche una serie di accessori, fondamentali per chi sviluppa applicazioni, come il generatore di help interattivi e il generatore della procedura di setup, alla Windows, della applicazione sviluppata. Quest'ultimo accessorio funziona con il sistema del Wizard.

Nel sistema è compresa la documentazione, accessibile sotto forma di help per Windows. Quindi tale documentazione è un ipertesto e, se serve, è anche copiabile, con il normale cut and paste. In particolare risultano copiables gli esempi di programmi.

Il neologismo nato con Visual Basic e con Visual C++ è «visual programming». Significa, in parole povere, che l'attività di programmazione, che un tempo consisteva nello scrivere lunghi e complessi listati, è diventata un'attività visuale, grafica più simile al disegno, in cui si tracciano oggetti sul video e se ne definiscono le caratteristiche in speciali pannelli, che li raccolgono. La scrittura delle istruzioni è limitata alle subroutine legate a ciascun oggetto dell'applicazione.

### Un piccolo esempio riassuntivo

MC, come detto, ha già parlato almeno in una mezza dozzina di articoli di Visual Basic. Il nostro problema, nel presentare la nuova versione di Visual Basic, è se rivolgerci a chi già lo conosce, in una delle precedenti versioni, magari anche perché ha letto i citati ar-

### Microsoft Visual Basic 3.0

#### Produttore e distributore:

Microsoft S.p.A.  
Centro Direz. Milano Oltre Pal. Tiepolo  
Via Cassanese, 224 - 20090 Segrate (MI)  
Tel.: 02/269121

#### Prezzi indicativi (IVA esclusa):

Visual Basic 3.0 Professional (inglese)	L. 690.000
Visual Basic 3.0 Standard (inglese)	L. 270.000

ticoli, oppure se ricominciare tutto da capo, ripetendo concetti però già ben noti al primo gruppo di lettori.

Cerchiamo di accontentare gli uni e gli altri presentando un semplice esempio pratico riassuntivo dei concetti generali e degli strumenti principali di Visual Basic.

Quando si lancia Visual Basic appare una barra con il menu. Possono apparire, nel senso che sono opzionali, una serie di altri elementi. La toolbar, con i bottoni scorciatoia per le varie operazioni concernenti la programmazione, la toolbox, che racchiude i vari elementi prefabbricati utilizzabili. Si tratta dei cosiddetti control. Alcuni di questi sono residenti, altri possono essere caricati o scaricati a seconda che servano o meno. Nella successiva figura 7 vediamo l'aspetto della toolbox prima e dopo lo scaricamento dei control di VB3 non residenti.

Al lancio appare anche una form, una finestra sulla quale si comincia a costruire l'applicazione, portandoci sopra degli oggetti. Nelle figure 1 e 2 vediamo una form, chiamata prova di Visual Basic 3.0, con un timer (che in fase di esecuzione non si vede), alcuni

testi (TextBox), riquadrati o meno, due option box (pari o dispari, se è pari non può essere dispari), due scroll bar, una checkbox, che mostra la crocetta se il numero superiore è pari, una listbox, che mostra i nomi di una serie di personaggi Disney. Infine c'è un bottone che mostra una freccia.

Tutti questi oggetti possono essere disegnati, nel senso che vanno prelevati dalla toolbox e posizionati nella form e, se del caso, dimensionati e allineati alla griglia.

Ogni oggetto ha alcune caratteristiche che possono essere definite nella finestra delle proprietà oppure gestite con la programmazione. Ad esempio vanno definite tutte le proprietà estetiche delle scritte, degli oggetti grafici, ecc.

La programmazione vera e propria è spezzettata nel senso che una subroutine corrisponde ad un evento che accade su un oggetto.

Nel nostro esempio abbiamo:

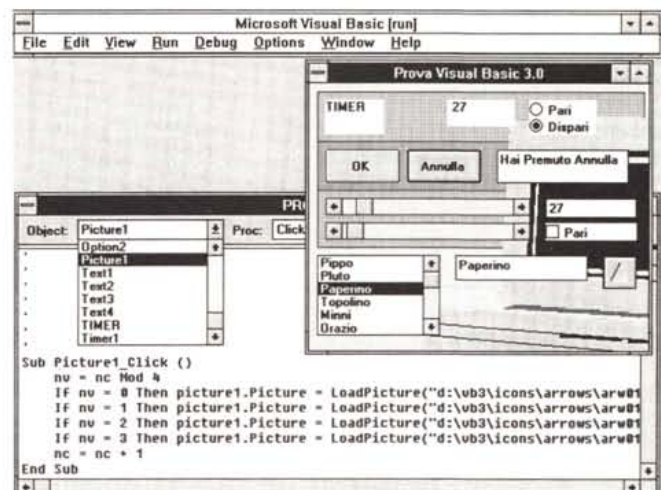
- l'evento timer. La temporizzazione prevede che ogni secondo venga aumentato di una unità il numero a fianco. Poi se tale numero è pari si accende l'OptionBox pari, altrimenti si spegne e si accende quella dispari. È il Control che provvede a tenere acceso una sola delle OptionBox;

- due bottoni. Un evento click su uno dei bottoni manda nella attigua TextBox una scritta;

- per le due scroll bar sono stati definite le proprietà minimo, pari a 0, e massimo, pari a 100. La TextBox mostra la moltiplicazione dei due numeri e la CheckBox viene attivata se tale risultato è Pari;

- la ListBox, alla Disney, viene caricata al verificarsi dell'evento load della form. La TextBox a fianco viene riempita con il nome del personaggio scelto;

Figura 1 - Visual Basic 3.0 - Esempio Pratico riassuntivo - Lato utente. MC ha più volte parlato del Visual Basic, nelle prove delle versioni precedenti ed in una serie di articoli organizzati in una specie di piccolo corso «elementare», dedicato agli utenti finali più che ai programmatori. A beneficio di chi si fosse perso queste puntate precedenti mostriamo un esercizio «riassuntivo» degli elementi fondamentali gestiti da Visual Basic. Tali elementi sono: Objeto, Proprietà, Metodi, Eventi, Istruzioni e Funzioni.



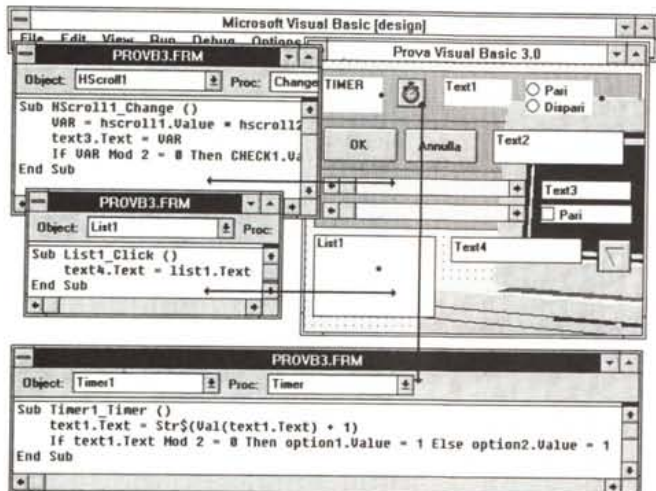


Figura 2 - Visual Basic 3.0 - Esempio Pratico riassuntivo - Lato programmatore. L'utente vede l'oggetto Form. All'interno dell'oggetto Form ci sono altri oggetti Windows. Ognuno degli oggetti ha una serie di proprietà. Ad un certo punto accade un Evento, il classico è il Click su un Bottone o su una voce di Menu. A tale evento si associa una Subroutine, che in generale esegue una serie di Istruzioni, che possono anche utilizzare delle Funzioni, ad esempio di tipo Matematico. La Subroutine produce la variazione di una o più proprietà di un Oggetto. Qualsiasi applicazione è composta di Oggetti e tutto quello che succede in una applicazione può essere ricondotto ad una serie di Eventi che generano la modifica delle Proprietà degli Oggetti.

Figura 3 - Visual Basic 3.0 - Generatore di Menu. Esiste un Editor che serve per scrivere le varie voci del Menu, che vanno opportunamente strutturate in livelli. Una volta scritte le voci nella finestrella dell'Editor il menu già appare e funziona in cima alla Form cui appartiene. Occorre poi, ovviamente, collegare all'evento Click sulla voce di Menu il «pezzo» di programma da eseguire quando si sceglie quella voce.

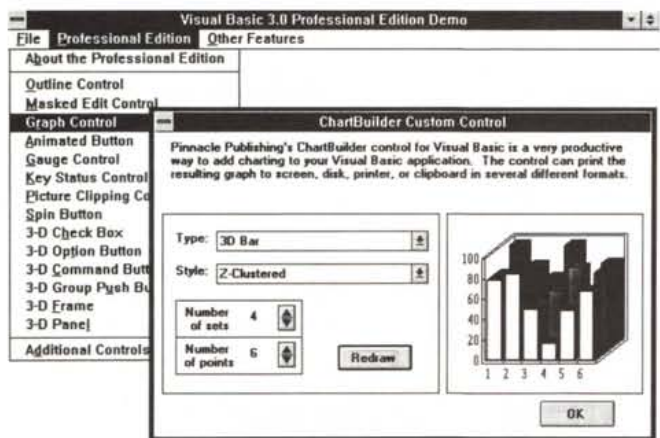
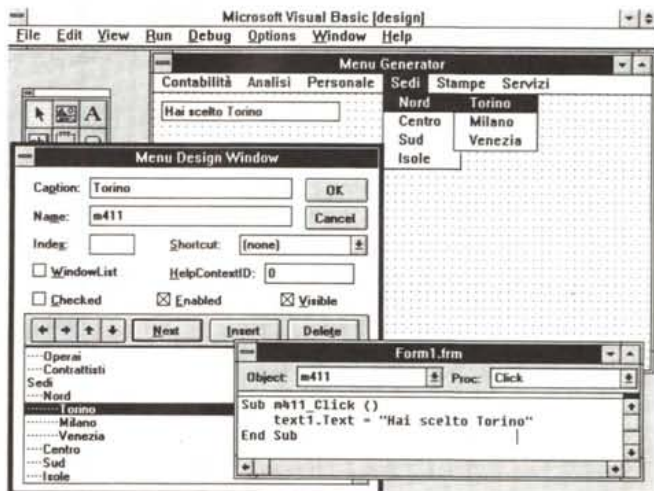


Figura 4 - Visual Basic 3.0 - Generatore di Grafici. Si tratta di un «motore» per generare Grafici, applicabile a qualsiasi applicazione. È molto sofisticato in quanto dispone di tredici tipi e di una cinquantina, in totale, di sottotipi. Gestisce inoltre tutti gli elementi tipici del Grafico: Titoli, Legende, Etichette, ecc. Si tratta di uno strumento versatile, direttamente utilizzabile nell'applicazione che si vuole sviluppare, in quanto è facilissimo mettere in relazione i propri dati con le Serie Numeriche graficate.

zioni, una o più proprietà degli oggetti. Una volta intuito questo meccanismo assolutamente elementare, si è capita concettualmente la programmazione Visual, e conseguentemente la programmazione Object Oriented.

### La manualistica

È ovvio che Visual Basic 3.0 è una evoluzione delle due versioni precedenti. I manuali essendo destinati anche ad eventuali «neofiti» comprendono anche tutti gli argomenti già presenti nei manuali delle versioni precedenti.

La conseguenza di questa stratificazione di versioni è una manualistica voluminosissima (conteremo circa 2.500 pagine), un vero record per tale tipo di prodotti.

Procediamo con ordine:

- Programmers Guide, 710 pagine. Tratta prima gli argomenti generali e poi individualmente i vari control. Seguono alcuni capitoli dedicati ad argomenti più tecnici come DDE, OLE, uso delle DLL, uso del Setup Wizard. Concludono alcune appendici tecniche.

- Language Reference. 675 pagine (e fanno 1385) in cui vengono messi in ordine alfabetico Oggetti, Funzioni, Istruzioni, Metodi, Proprietà e Eventi. Il primo argomento è la funzione matematica ABS e l'ultimo il metodo ZOrder, che indica la priorità dell'oggetto

- il bottone con la freccia viene modificato, appare una freccia in un altro senso, quando ci si fa click sopra.

Nella figura 1 vediamo oltre alla form anche il programma, si tratta di una subroutine legata all'evento click sull'oggetto Picture1. Al verificarsi dell'evento click viene modificata la proprietà picture dell'oggetto Picture1.

Nella figura 2 vediamo un collage di subroutine, con delle frecce che le legano agli oggetti da cui dipendono. La form viene vista in fase di lavorazione.

In figura 3 (anche questa fa parte del riassunto) vediamo invece all'opera il generatore di menu. Tali menu, che si

scrivono con una modalità che più facile di così non si può in questa speciale Dialog Box, vanno a finire direttamente nella form di appartenenza. Ogni voce di menu è un oggetto che può avere le sue caratteristiche (ad esempio può essere disabilitata) e che può subire l'evento click.

In definitiva gli elementi che si manipolano sono oggetti, proprietà, istruzioni, funzioni, metodi, eventi.

Un'applicazione per quanto complessa sia è costituita da una serie di oggetti, che hanno delle proprietà. Il programma, al verificarsi di un evento, modifica, attraverso una serie di istru-

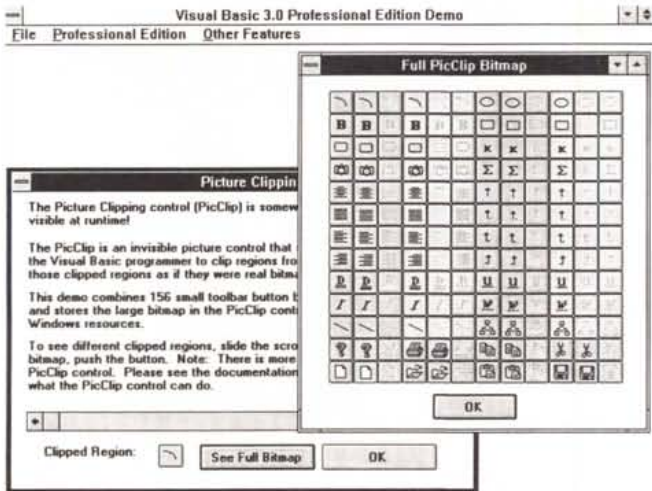
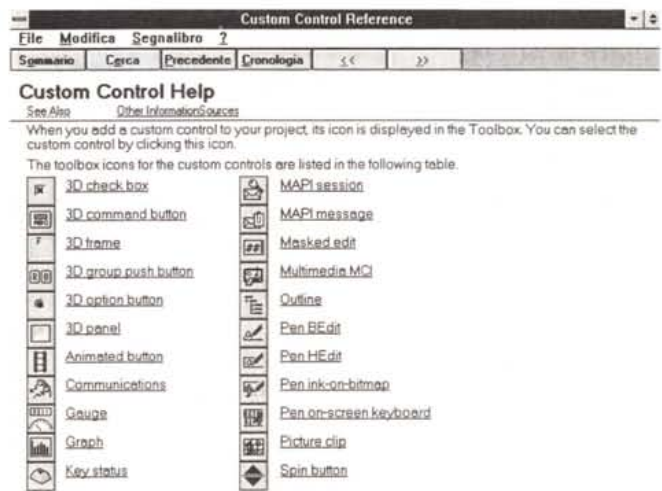


Figura 5 - Visual Basic 3.0 - Strumento ClipBit. Con questa figura vogliamo esemplificare il modo con cui, da Visual Basic, viene facilitato lo sviluppo di nuove applicazioni usando elementi preconfezionati. Con lo strumento ClipBit è possibile in pratica definire una Matrice di Bottoni, che poi possono essere facilmente richiamati con il loro indice numerico.

Figura 6 - Visual Basic 3.0 - Help Relativo al Primo Strato di Controls. Poiché i Controls aggiuntivi possono essere caricati o scaricati a seconda delle necessità, i progettisti sono stati costretti a fornire due file di help separati, il che in certi casi comporta l'inconveniente di dover passare dall'uno all'altro. Inoltre il secondo, quello dei Custom Controls, di cui vediamo la prima pagina, con l'indice dei Controls aggiuntivi, è meno ricco del primo, in quanto mancano del tutto i comodissimi e copiabilissimi esempi.



nella form. Tale priorità si varia con i comandi front e back.

- Professional Features Book 1, che contiene tre parti distinte e separate. Le sezioni Reference relative ai Control Custom (quelli in più rispetto alla dotazione di base). La sezione più tecnica riservata a chi vuol programmare, magari usando Visual C++, ulteriori Controls per Visual Basic. Infine la sezione relativa all'Help Compiler Guide. Contiamo le pagine. Stavamo a 1385, cui aggiungiamo le tre sezioni numerate individualmente, di 365, 330 e 190 pagine. Totale 2170.

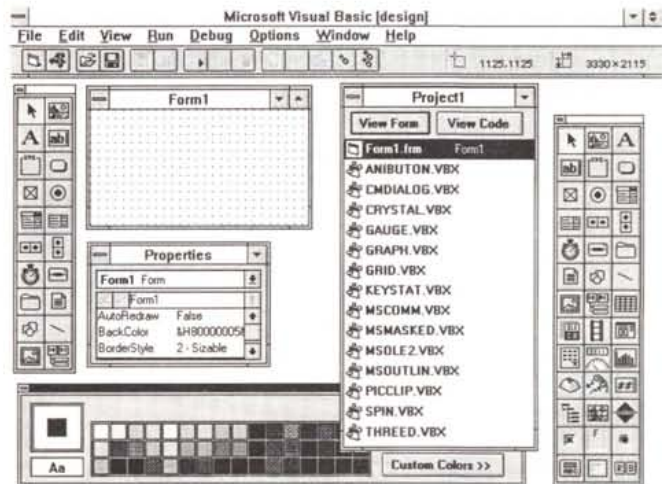
- Professional Features Book 2, che contiene tutto ciò che riguarda il Control più interessante che è il Data Control. La sezione si chiama Data Access Guide ed ha 155 pagine. C'è poi il manuale di un accessorio in dotazione, che si chiama Crystal Report, e che serve per generare dei report in formato Windows partendo da file in vari formati esterni.

Il manuale del Crystal Report, che è impaginato in tutt'altra maniera anche se è rilegato nel Book 2, ha circa 130 pagine.

Il totale raggiunge quindi le 2465 pagine.

Nelle prime pagine dei manuali sono state inserite delle comode tabelle riassuntive in cui vengono incrociati oggetti e proprietà, eventi e oggetti, metodi

Figura 7 - Visual Basic 3.0 - Strip-Tease integrale dei Controls. La versione 3.0 di Visual Basic Professional contiene una stratificazione di Control. Ci sono quelli fondamentali, già presenti nella versione 1.0, più vari altri, alcuni già visti nei successivi aggiornamenti, alcuni «inediti». In questa figura, si tratta di un collage, vediamo le due Toolbox, quella di base e quella con tutti i Control aggiuntivi caricati a bordo. Si può notare come sia possibile caricare o scaricare, a seconda che serva o meno, il Control.



ed oggetti, e delle tabelle in cui sono categorizzate le funzioni e le istruzioni di programmazione.

Interessanti sono le tabelle riferite all'oggetto database, che è la vera novità di questa nuova versione di Visual Basic.

I dischetti sono 9, e se si sceglie di installare tutto, programmi principali e materiale accessorio, lo spazio richiesto su hard disk è di 32 megabyte.

### Piccole novità

Come detto il Visual Basic è modulare nel senso che è possibile aggiunge-

re dei control, o di mercato, sviluppati cioè da software house, oppure sviluppati in proprio utilizzando magari il Visual C++ della stessa Microsoft (una parte del manuale è dedicata a tale attività).

Sin dalla versione 1.0, con il Professional Toolkit, la Microsoft ha ufficializzato control sviluppati da terzi. Questo significa che vengono inglobati e certificati «moduli» che si aggiungono al modulo base.

Molte delle novità aggiunte via via sono proprio di questo tipo. Ad esempio il Control Graph (fig. 4) è opera dalla Pinnacle Publishing Inc. Si tratta di



Figura 8 - Visual Basic 3.0 - Il pioniere dell' OLE 2.0. Uno degli aspetti tecnologici più evoluti che caratterizzano il sistema Windows è l'Object Linking and Embedding, il famigerato OLE. È stato recentemente definito lo standard OLE 2.0, di cui potremo più compiutamente parlare quando saranno molti i prodotti che lo adottano. In questo esempio, preso tra i numerosissimi disponibili nel materiale VB3, la piccola calcolatrice (è una applicazione Windows a tutti gli effetti) viene utilizzata direttamente dall'applicazione VB3, che la vede come proprio accessorio pur essendo un'applicazione a sé stante.

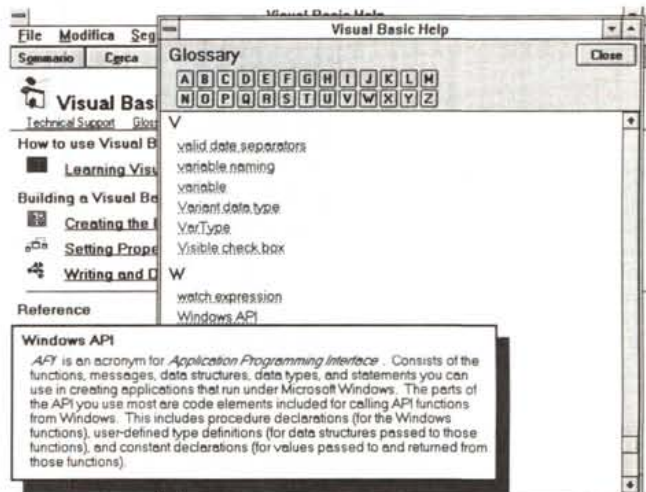


Figura 9 - Visual Basic 3.0 - Glossario Help. Il «taglio» per utente del Visual Basic è verificabile anche dagli Help, che sono molto didattici, nel senso che spiegano anche il significato teorico dei vari comandi o delle varie istruzioni. In tal senso è molto utile e significativo il Glossario, sotto forma di Help, con la comoda pulsantiera a lettere, che permette di eseguire delle ricerche alfabetiche direttamente con il Mouse.

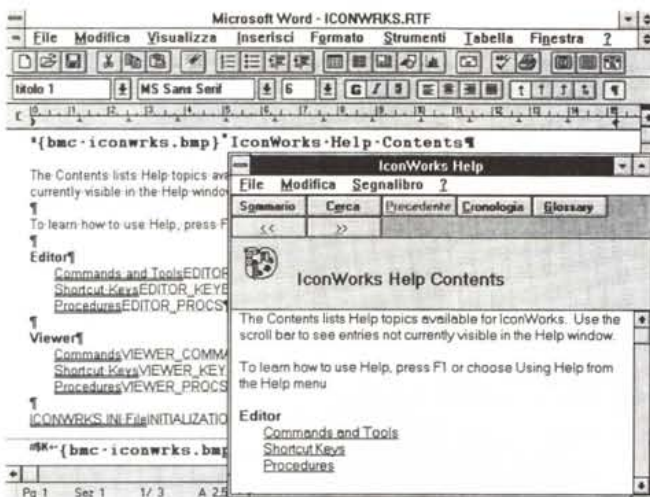


Figura 10 - Visual Basic 3.0 - Help Generator. Nel Sistema VB sono compresi anche una serie di accessori, fondamentali per chi sviluppa applicazioni, come il generatore di Help interattivi, il generatore della procedura di Setup. Per realizzare l'Help occorre partire da un testo scritto con un Word Processor. Nel testo vanno rispettate alcune caratteristiche compositive e vanno inserite delle codifiche speciali. Il testo poi va «compilato» con un programma in dotazione. Il risultato è un file HLP, consultabile attraverso il WINHELP.EXE di Windows.

di realizzare applicazioni integrate con il sistema di posta elettronica.

Questo a conferma della versatilità del Visual Basic soprattutto in quelle applicazioni più spinte tecnologicamente (nel senso che debbono sfruttare al massimo l'hardware o debbono sfruttare un hardware particolare), ma meno critiche in termini di performance.

Visual Basic 3.0 è OLE 2.0 compatibile. Nella documentazione c'è un esempio di applicazione OLE 2.0 che però si appoggia su un'applicazione Calcolatrice, messa lì come «sparing partner» (fig. 8).

Se permettete parleremo in un successivo articolo di OLE 2.0, quando esisterà un numero sufficiente di applicativi OLE 2.0 compatibili, che permettano di sperimentare tutte le casistiche di trasferimento dati via OLE.

un vero e proprio motore che genera moltissimi tipi di grafici (ci sono anche quelli 3D, quelli polari, ecc.) e che può essere inserito in qualsiasi applicazione.

Citiamo anche il Control ClipBit, che è più una utility, nel senso che serve un po' da per tutto in tutte le applicazioni (fig. 5). Con ClipBit è possibile definire una unica array di bottoni, dalla quale richiamare facilmente, tramite il relativo indice numerico, quello che serve in quel momento nella procedura.

Nella figura 6 vediamo il Summary (in pratica l'indice) che mostra tutti i

nuovi control aggiuntivi a quelli di base. Nella figura 7 (si tratta di un collage) invece vediamo le due situazioni, quella con tutti i control a bordo (si aggiungono alla Toolbox) e quella con i soli Controls di base.

Esistono Controls con finalità più estetiche, ad esempio tutti quelli che servono a gestire oggetti con look 3D, Controls più sofisticati, come il gestore di Outline, oggetti organizzati gerarchicamente, oppure come quello che serve per gestire comunicazioni tramite la porta seriale, oppure quello che consente di sfruttare le librerie MAPI (Messaging API) e che quindi consente

### Due accessori indispensabili l'Help Compiler e il generatore di dischetti di setup

Prima di passare a parlare dell'unica vera grande novità presente nel Visual Basic 3.0, il Data Control, citiamo la presenza nel materiale innanzitutto della documentazione accessibile anche al di fuori del VB. Il «taglio» è quasi divulgativo. Si veda ad esempio il glossario che spiega i vari termini in cui molto spesso ci si imbatte (fig. 9).

Purtroppo, trattandosi di un prodotto ormai modulare, l'help è stato spezzato

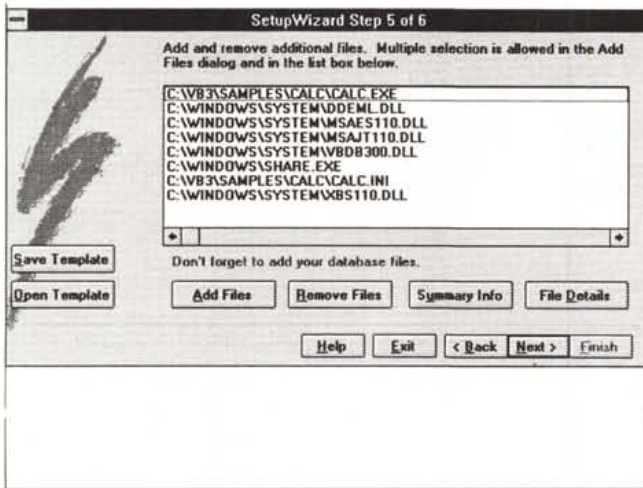
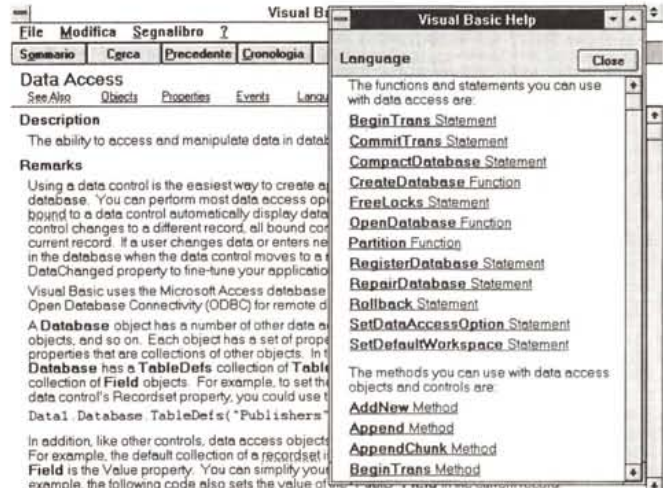


Figura 11 - Visual Basic 3.0 - Setup Wizard. Una volta realizzata la propria applicazione va prodotto un dischetto, o i dischetti, per il Setup, che deve rispettare le regole di tutti i Setup di tutti gli Applicativi di Windows, anche quelli importanti. La generazione si avvale di un comodo Wizard e permette di inserire tutti i file che si vuole, ad esempio i file con i Dati o i file di Esempio, e di compattare i file nei dischetti.

Figura 12 - Visual Basic 3.0 - Data Access - L'Help Specifico. La più interessante novità presente nel Visual Basic 3.0 consiste nel Data Control, che consente di collegare la applicazione VB con un Database esterno, che viene visto come un «oggetto». Tutte le operazioni di Database, definizione, aggiornamento, manipolazione, ecc. possono essere eseguite da Visual Basic, senza dover ricorrere alla programmazione.



in più parti. Inoltre, ancora purtroppo, le sezioni relative ai nuovi Controls, non sono dotate di quei bellissimi, e copiabilissimi, esempi di cui è dotata la prima parte, quella relativa ai Controls di base.

C'è da citare la significativa presenza dell'Help Compiler (si tratta della versione 3.1, già presente in VB 2.0) che serve agli sviluppatori per realizzare e per collegare alla propria applicazione un help «alla Windows», gestito, come qualsiasi help di Windows, dal WINHELP.EXE di Windows.

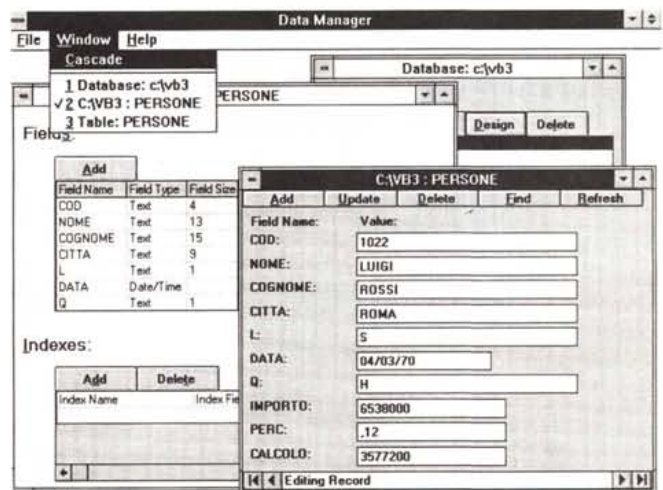
Come noto l'help di Windows è un ipertesto. Si può ipotizzare un utilizzo dell'help di Windows come gestore di documenti ipertestuali e quindi un utilizzo indipendente da qualsiasi applicazione (fig. 10). Se permettete ne parliamo un'altra volta.

Altro accessorio indispensabile per chi realizza e deve distribuire applicazioni Windows è il generatore di procedure di setup. La generazione è guidata da un Wizard (procedura guidata passo passo, fig. 11). Le varie dialog box coinvolte possono essere delle normali form di Visual Basic.

### Una grossissima novità. L'accesso ai Dati

La novità più importante, che giustifica la nascita della versione 3.0 del Visual Basic, consiste sicuramente nelle nuove e sofisticate modalità di accesso

Figura 13 - Visual Basic 3.0 - Opzione di Menu Data Manager. Nel menu Option del menu principale c'è la voce Data Manager. In pratica viene richiamata un'applicazione che permette l'accesso alle Basi Dati esterne, nei formati riconosciuti dallo strumento Data Control. Attraverso tale programma è anche possibile definire la struttura della Banca Dati e le relative Tabelle.



ai dati, che si aggiungono a quelle, più rudimentali, precedentemente presenti.

Il Basic, qualsiasi versione, è sempre stato un po'... debole in trattamento dati. Non è mai stato un linguaggio adatto allo sviluppo di applicazioni gestionali, realizzate preferibilmente con prodotti tipo DBMS.

Il Visual Basic essendo, come detto, modulare, si presta ad essere coadiuvato da control specializzati nella gestione dei database. Molti di questi sono stati prodotti da software house indipendenti, proprio allo scopo di permettere al Visual Basic di dialogare con

database esterni, simulandone tutte le funzioni. Usando tali Controls in pratica dal Visual Basic si possono definire strutture, impostare indici, alimentare, modificare, cancellare record di tabelle.

L'argomento VB e DBMS è molto vasto in quanto riguarda senza dubbio i metodi precedenti, come quello classico nel Basic (trattato nell'articolo del numero 126 di MC), oppure quello costituito dall'accesso a dati esterno tramite driver ODBC.

La novità principale della versione 3.0 è costituita dalla presenza di un nuovo control, che si chiama Data Control, che permette di creare un aggan-

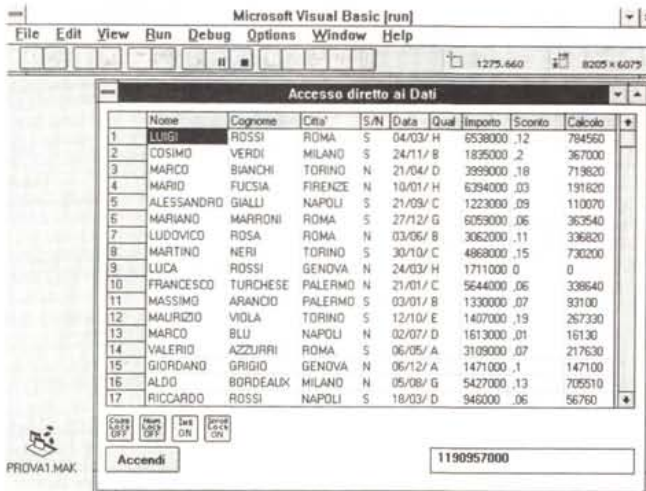


Figura 14 - Visual Basic 3.0 - DataControl, Grid, Out. Ecco un nostro primo esperimento di Accesso ai Dati. Abbiamo creato una Grid ricevente, in cui riversiamo una serie di dati provenienti da un semplice archivio DBF. In basso Cinque Bottoni, che non c'entrano nulla con i dati, che indicano la posizione dei Tasti BlocNum, CapsLock, Ins e BlocScor e sono gestiti da un ulteriore Control. Il Bottone Accendi/Spigni serve per attivare o disattivare tali interruttori.

Figura 15 - Visual Basic 3.0 - DataControl, Grid, ecc. Dietro le quinte. Vediamo come si presenta il programma legato all'evento Form Load, che è quello che definisce, come prima cosa, i titoli da inserire nella prima riga della Griglia. Poi, con una classica istruzione DO WHILE, viene alimentata la Griglia con i dati letti dall'archivio esterno. Nella prima colonna viene inserito il progressivo di riga e nell'ultima un contenuto calcolato. Da notare la semplicità delle istruzioni che consentono di definire e di usare il Database esterno.

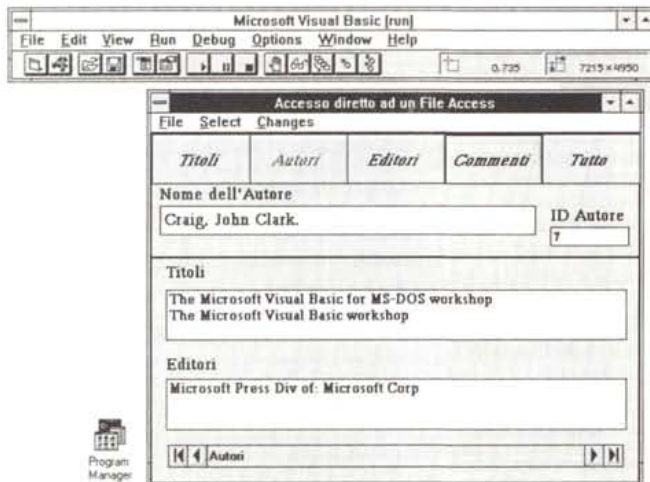
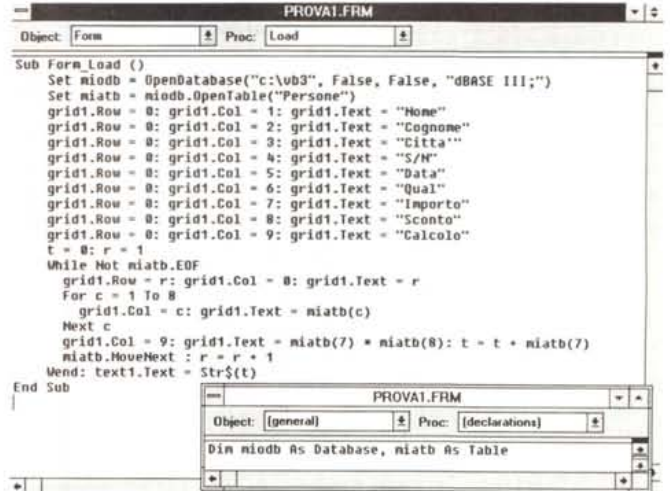


Figura 16 - Visual Basic 3.0 - Esempio Data Control - Out. Tra il materiale esemplificativo fornito con Visual Basic 3.0 c'è questo esempio di uso «evoluto» del Data Control. Evoluto nel senso che viene «visto» un intero Database Access, composto da quattro Tabelle e da varie Query. Si tratta di una applicazione Biblioteca, accessibile dalla parte del Titolo, dell'Autore e dell'Editore. Nella figura una Schermata «Iato autore» in cui c'è la Lista dei Volumi, che sono in relazione uno a molti con l'Autore.

che, a parità di funzionalità, vede migliorate le prestazioni, e vede aumentati gli accessori. In particolare è disponibile un runtime, che permette di distribuire l'applicazione Access ad utilizzatori che non hanno Access, e sono disponibili accessori per generare gli help e per generare i dischetti di setup. Va anche detto che se Access è il formato «nativo» dei database di Visual Basic, lo stesso non disdegna l'utilizzo di altri formati, xBase (dBase III, dBase IV, FoxBase, ecc.) e Paradox in testa.

### L'oggetto Database

Non abbiamo lo spazio necessario per approfondire il discorso. Lo faremo in un prossimo articolo (già in preparazione). Per ora vi diamo un'indicazione di massima di come si interfaccia un database esterno con un form disegnato con Visual Basic 3.0 e vi presentiamo una serie di immagini riepilogative di alcuni possibili utilizzi del Visual Basic con database esterni.

Supponiamo di disporre di un semplicissimo database access chiamato DATI, in cui esista una tabella chiamata TABE, in cui esistano due campi, uno alfanumerico chiamato NOME, e uno numerico chiamato NUMERO.

Occorre creare una form in cui:  
 - inserire un Data Control:  
 - il cui Name sia NOMED (lo scegliamo noi)

cio con un database esterno. Una volta agganciato il database esterno, è possibile collegare gli altri control ai dati del database.

Ma l'accesso può all'occorrenza essere realizzato «saltando» il Data Control, sfruttando direttamente le nuove istruzioni Visual Basic che permettono di definire un «oggetto» database e tutte quelle conseguenti che permettono di manipolarlo.

Altre novità sono costituite dagli ulteriori accessori che rendono possibile sia gestire totalmente dal Visual Basic la definizione del database (esiste

all'uopo il programma accessorio Data Manager), sia di inserire, nella propria applicazione, dei report realizzati tramite il programma Crystal Report.

Per quanto riguarda i formati di database va detto che quello «nativo» del Visual Basic è il formato Access.

In altre parole Access va considerato sia un prodotto (Microsoft Access) sia un Formato Database, indipendente dal prodotto Access, utilizzabile in tutte le sue funzioni anche da altri prodotti, Visual Basic in testa.

Va detto con l'occasione che è anche uscita la versione 1.1 di Access,

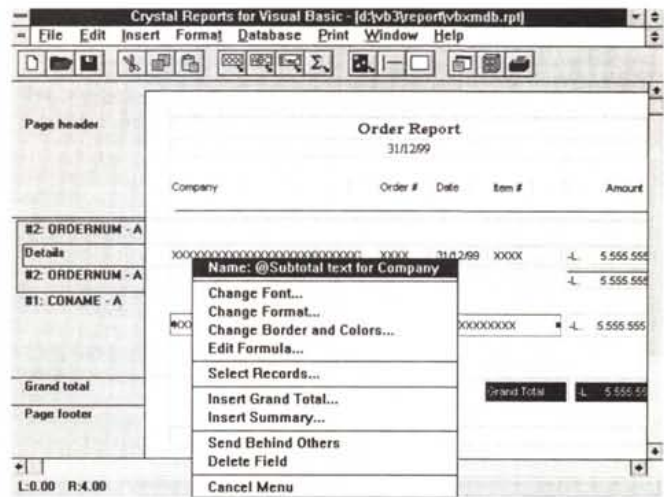
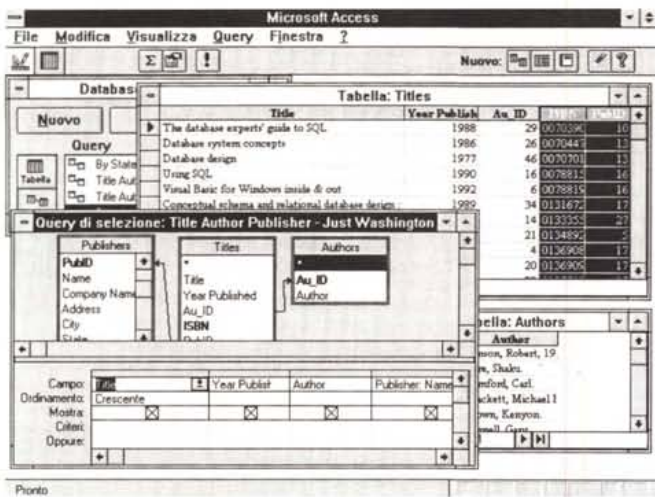


Figura 17 - Visual Basic 3.0 - Esempio Data Control - Lato Access. Visual Basic 3.0 ha un rapporto privilegiato con Access, anzi il formato Dati «nativo» di Visual Basic è proprio Access, che per essere ben sfruttato necessita di una buona conoscenza di questo secondo prodotto. Qui vediamo il Database Biblioteca direttamente da Access. Notiamo che contiene non solo Table ma anche delle Query anche queste sfruttabili direttamente da VB3.

Figura 18 - Crystal Report for Visual Basic - Struttura e tasto Destro. Crystal Report per Visual Basic... non ha molto a che vedere con Visual Basic. Si tratta di un'applicazione chiusa, a sé stante, che serve per generare dei Report, anche di una certa complessità, partendo da file di dati di qualsiasi genere. Probabilmente è stato inserito nel pacchetto per poter affiancare le funzionalità di DBMS che nel Visual Basic 3.0, grazie al Control Data e all'allineamento alle specifiche ODBC, sono di molto potenziate.

- la cui proprietà database sia DATI
- la cui proprietà Record Sources sia TABLE
- inserire una Text Box:
- la cui proprietà Datasource sia NOMED
- la cui proprietà Datafield sia NOME
- inserire una MaskEdit (una TextBox con un formato numerico incorporato):
- la cui proprietà Datasource sia NOMED
- la cui proprietà Datafield sia NUMERO
- la cui proprietà Format sia #.##0.

Il Data Control appare come una specie di scroll bar, con due coppie di frecce ai due estremi (lo vediamo in basso in figura 16).

Eseguite questa dozzina di operazioni il collegamento è realizzato, per cui lanciando il comando RUN, la form diventa attiva e il Data Control permette di scorrere i dati.

Va detto che i dati sono anche modificabili (se si vuole).

Se si vuole la totale gestione dei dati si possono sfruttare una serie di istruzioni ora attive per il fatto che il database è stato attivato.

- Nel nostro caso alcune istruzioni utilizzabili potrebbero essere:
- NOMED.Recordset.MoveNext
  - NOMED.Recordset.MovePrevious
  - NOMED.Recordset.MoveFirst
  - NOMED.Recordset.MoveLast

(equivalenti alle quattro frecce presenti nel Control)

- NOMED.Recordset.Addnew
  - NOMED.Recordset.Update
  - NOMED.Recordset.Delete
- a livello di Record. Ma ce ne sono anche a livello di Field. Ad esempio:
- NOMED.Recordset(«NOME») = «PIPP0»
  - NOMED.Recordset(«NUMERO») = 6000.

Quindi i database, le tabelle, i campi delle tabelle, gli indici delle tabelle sono visti come «oggetti». Esistono istruzioni che agiscono su tali oggetti (Data Manipulation Language), esistono controlli che possono scambiare dati con tali oggetti, ecc.

È inoltre chiaro che è possibile accedere anche agli altri oggetti del database Access, prime tra tutte le query.

È chiaro che vanno riconsiderati tutti gli aspetti relazionali, che Access risolve... gratuitamente, mentre Visual Basic 3.0 un po' meno.

È chiaro che è possibile accedere ad altri tipi di database.

È chiaro che è possibile sia utilizzare database creati al di fuori, sia creare il database direttamente da dentro Visual Basic, sia infine creare il database usando un accessorio all'uopo messo come dotazione di Visual Basic 3.0.

L'argomento è nuovo ed è soprattutto enorme per cui lo approfondiremo, come detto, in un prossimo articolo.

Con l'occasione precisiamo che la più grossa differenza tra la versione

Professional, quella provata, e quella normale, di Visual Basic 3.0, sta nel fatto che in quella normale non sono disponibili le istruzioni di generazione del database esterno.

Passiamo alle figure, dalla 12 alla 19, che non commentiamo in quanto sono dotate di ricche didascalie.

### Conclusione

Visual Basic si va consolidando nel senso che le nuove versioni da una parte confermano tutto quanto era presente nelle versioni precedenti, e confortano chi ha creduto subito in questo nuovo strumento, dall'altra, apportando nuovi Controls, ne allargano ulteriormente le possibilità applicative, i cui confini coincidono ormai in pratica con quelli di Windows.

Con Visual Basic si possono sviluppare applicazioni contabili, scientifiche, grafiche, di posta elettronica, applicazioni di servizio, ecc. Insomma applicazioni di tutti i tipi.

La novità più rivoluzionaria presente nel Visual Basic 3.0 è il Control Data, con il quale Visual Basic si candida ad essere anche un sistema di programmazione adatto per applicazioni gestionali di grande impegno, ma di questo, come detto, parleremo nel prossimo numero.