

Borland dBASE IV 2.0 & dBASE Compiler 1.0

di Valter Di Dio

La forza del dBase, fin dalla sua prima comparsa nel lontano 1981, consiste nell'aver portato la semplicità di un programma scritto in Basic nella gestione degli archivi e, soprattutto, degli indici. E, come il Basic, il dBase era, fino ad oggi, un linguaggio interpretato; quindi comodissimo per il debug ma lento nell'esecuzione e potenzialmente pericoloso in mano ad utenti poco esperti. Il problema fu prontamente risolto da altre ditte che hanno provveduto a colmare le principali lacune del dBase, ad esempio la Nantuket con il suo compilatore Clipper Summer 87, finendo però nello stesso tempo per consacrare come standard il formato DBF. Il Clipper, il FoxBase o altri prodotti simili, pur non essendo compatibili al 100% con il dBase III, permettevano di sviluppare gran parte dell'applicazione in modo interpretato e alla fine ritoc-

care e compilare il tutto presentando in uscita un programma eseguibile autonomo e veloce. Con l'uscita nel 1988 del dBase IV versione 1.0, e le successive versioni 1.1 e 1.5, quest'ultima dopo il passaggio da Ashton Tate a Borland, sono state eliminate le principali limitazioni del linguaggio, come ad esempio la mancanza degli array e di una gestione automatica di finestre e menu, ma la velocità ne ha risentito parecchio, al punto da rendere quasi indispensabile l'uso di un compilatore.

Il dBase IV possedeva già dalla prima versione la gestione della multiutenza, delle transazioni, dei menu a tendina e pop-up e delle finestre, in più, con un ottimo sviluppatore di applicazioni e il debugger, dimostrava di avere tutte le carte in regola per diventare il pacchetto ideale per uno sviluppatore professionale. Anche nei confronti dell'utente

con scarse attitudini a programmare il dBase IV si presentava sicuramente più utilizzabile grazie al nuovo Control Center (che sostituiva in pratica il vecchio Assist) e, soprattutto, grazie al SQL (Structured Query Language) integrato e in grado di permettere un semplicissimo Query By Example.

Però, se per l'utente finale, a parte i problemi di velocità, il pacchetto poteva anche andare bene così, lo stesso non si poteva dire degli sviluppatori che necessitavano assolutamente di un compilatore che da un lato non imponga all'utente l'acquisto del pacchetto o delle licenze, dall'altro impedisca la modifica, intenzionale o casuale, del software consegnato. E infatti il compilatore era stato promesso sin dalla prima uscita del dBase IV ma, a distanza di oltre quattro anni, la comunità degli sviluppatori aspettava ancora il compilatore e

l'attesa sembrava diventare sempre più lunga. Alcune voci maligne cominciavano ad insinuare che la Borland, possedendo già un potentissimo database relazionale (il Paradox) non avesse alcun interesse a investire nel dBase IV invece ecco la più tangibile delle smentite: una nuova velocissima versione di dBase e il relativo compilatore.

L'installazione

Il paragrafo sulla installazione, nel caso del dBase IV e del suo compilatore, si riassume nella semplice constatazione che è andato tutto bene al primo colpo. E questa è una cosa abbastanza rara al giorno d'oggi con le macchine sempre più diverse e sempre con qualche cosa personalizzata oltre il limite. Unica avvertenza, che comunque il programma di installazione comunica doverosamente, è la quantità di spazio libero sul disco necessaria a completare l'operazione. Il dBase IV, una volta installato, occupa sul disco rigido da un minimo di 4,5 megabyte, nella configurazione base, ad un massimo di 7,5 megabyte nella configurazione completa di tutti i programmi di esempi, il demo introduttivo e i tool di sviluppo. Durante la fase di installazione però ne servono ancora altri due necessari per i file temporanei durante il processo di espansione. Nel caso si usasse un disco rigido compresso con Stacker o con il nuovo Dos 6, il manuale di installazione suggerisce di controllare, tramite le utility apposite fornite con Stacker o con il Dos 6, che ci sia effettivamente lo spazio libero necessario o comunque che il comando Dir comunichi almeno 10 megabyte liberi; questo perché la effettiva capacità di un disco compresso è solo teorica e dipende molto dal tipo di dato che vi verrà scritto.

Nel caso si stia aggiornando una precedente versione di dBase IV si può istruire il programma di installazione affinché non tocchi i vecchi file di lavoro e di configurazione. L'ideale sarebbe comunque di lasciare la vecchia versione intatta e installare la nuova su una directory differente: non si sa mai.

L'installazione del compilatore è altrettanto semplice, richiede poco più di 4 megabyte di spazio libero sul disco durante la fase di espansione degli archivi compressi e occupa alla fine circa 2,8 megabyte. Anche per il compilatore valgono le avvertenze di prima nel caso che il disco sia compresso.

Il doppio Code-Page

Con la versione 2.0 di dBase è stata data la possibilità di stabilire il formato

Borland dBASE IV 2.0

Produttore e distributore:

Borland Italia S.r.l.
Centro Direzionale Milano Oltre
Pal. Leonardo - Via Cassanese, 224
20090 Segrate (MI)
Tel.: 02/269151

Prezzi (IVA esclusa):

Borland dBase IV 2.0	L. 1.390.000
Borland dBase IV 2.0 (aggiornamento)	L. 249.000
Compilatore	L. 699.000
Compilatore (aggiornamento)	L. 349.000

del Code-Page scegliendolo tra quello standard 437 e quello europeo 850. Il Code-Page indica quali caratteri speciali (ad esempio accentate, simboli, vocali con dieresi, ecc.) vengono assegnati alla tabella dei codici ASCII in corrispondenza dei valori maggiori di 127. Durante l'installazione il programma controlla quale tabella è stata utilizzata nel file Config.sys della macchina su cui si sta installando dBase e propone automaticamente il codice corretto. Il Code-Page accettato durante l'installazione viene scritto nel file di configurazione del dBase e verrà utilizzato per marcare tutti gli archivi DBF che verranno creati. I file DBF delle versioni precedenti, che non avevano Code-Page, verranno considerati come se fossero stati fatti con la tabella 850 (l'unica esistente in precedenza).

Il dBase IV nella versione 2.0 esegue un controllo, escludibile, sul tipo di Code-Page utilizzato nel database che si sta aprendo e se nota una differenza da quello impostato chiede di confermare l'esecuzione dell'operazione al fine di

evitare involontari danneggiamenti degli archivi, o degli indici, se questi contenessero caratteri il cui codice ASCII è maggiore di 127.

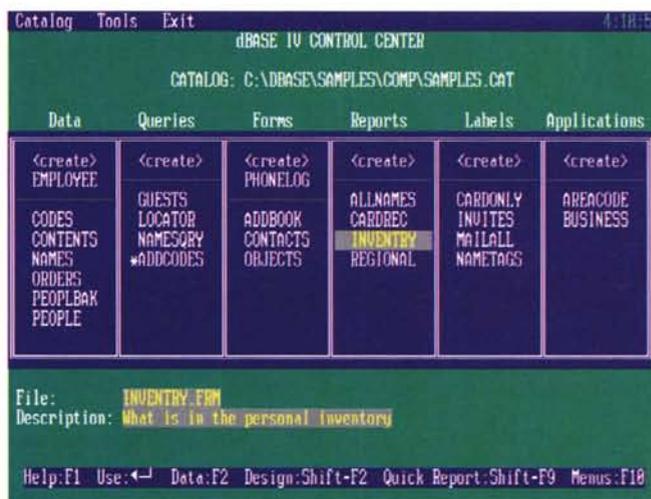
Anche se durante l'installazione si è scelto un determinato Code-Page è sempre possibile sostituirlo o, nel caso del compilatore, istruire dBase IV affinché usi un differente Code-Page. Purtroppo il programma risultante non sarà autoconfigurante, ma è necessario conoscere prima il Code-Page della macchina su cui dovrà girare. Nel caso si utilizzi un'applicazione dBase in rete locale, è indispensabile che tutte le workstation collegate alla rete abbiano lo stesso Code-Page.

Quale Hardware

La macchina ideale su cui far girare dBase IV è un 486 da almeno 20 MHz con minimo 4,5 megabyte di RAM. Il programma gira bene anche su macchine molto più limitate, ad esempio abbiamo compilato una applicazione che gira tranquillamente in rete su dei 286 diskless con solo un megabyte di RAM, ma in questo caso la velocità ne risente in modo sensibile e non può essere utilizzato il gestore di memoria virtuale del dBase. Stranamente anche sostituendo le macchine con dei 386 la velocità non migliora di molto, mentre passando sui 486 il salto è decisamente sensibile e si ottengono prestazioni quasi dieci volte più veloci. Questo strano comportamento è probabilmente dovuto più che altro alla presenza nei 486 della MMU e della cache che consentono un uso ottimale del disco e della memoria virtuale.

Se è previsto che l'applicazione debba lavorare in rete è preferibile avere un hard disk anche sulle workstation in modo che dBase IV possa installarci la configurazione di utente e alcuni degli

Il nuovo Control Center che sostituisce il vecchio Assist. Può essere richiamato anche da dentro Windows e fornisce uno strumento completo per lavorare in dBase senza conoscerne i comandi.





Borland dBase Compiler 1.0

Il dBase è nato come linguaggio interpretato, questo significa che, dentro al dBase IV, esiste una routine che analizza il file dei comandi (.prg) scritto dal programmatore, capisce, o cerca di capire, i comandi che ci sono scritti ed esegue le istruzioni ad una ad una.

In effetti è come se invece di un vero e proprio programma si fosse scritta una specie di grossa macro. Le limitazioni di un simile approccio sono la scarsa velocità di esecuzione, dovuta al fatto che se si esegue dieci volte la stessa riga in dBase ogni volta viene letta e interpretata nuovamente, e il fatto che se una parte del programma contiene errori, ma non viene mai eseguita durante le prove, è impossibile accorgersene. Il vantaggio sta in una struttura semplice del linguaggio e nella possibilità di interrompere in qualsiasi momento l'elaborazione, eseguire a mano dei comandi e poi riprendere tranquillamente dal punto in cui ci si era fermati.

In realtà questo era vero con il dBase III perché con la versione IV è stato introdotto un passo intermedio tra il linguaggio interpretato e quello compilato. Infatti il dBase IV prima di eseguire un programma per la prima volta, o

dopo una modifica, esegue una meta-compilazione (che purtroppo sia il manuale che la scritta a video chiamano compilazione). In questa fase il programma viene controllato sintatticamente e vengono predisposte le chiamate alle routine principali. Il file .PRG diventa allora .DBO ed è immediatamente eseguibile dando il comando Do da dentro al dBase IV.

È evidente quindi che per eseguire un programma scritto in dBase si deve possedere il pacchetto dBase o almeno una parte di esso detto RunTime.

Per contro un compilatore è un programma che genera, a partire dal file PRG scritto dall'utente, direttamente un codice eseguibile .EXE in cui tutte le istruzioni sono state sostituite dalle parti di codice che le eseguono, tutti i riferimenti a variabili, maschere, formati e subroutine sono stati verificati e realizzati e sono state incluse tutte le parti del RunTime necessarie all'esecuzione.

La compilazione di un programma permette quindi di aumentare la velocità di esecuzione perché la fase di interpretazione viene eseguita una volta per tutte e di verificare, sempre prima di generare il codice finale, che tutte le

parti di programma siano presenti e correttamente scritte.

Se vogliamo essere pignoli, il compilatore del dBase IV non è un vero e proprio compilatore perché, probabilmente per garantire la compatibilità al 100% con i programmi interpretati, non risolve completamente tutte le chiamate e non integra nel codice il RunTime, ma lo accoda ad esso o, volendo, non lo aggiunge nemmeno. Questa modalità di funzionamento ricorda molto i primi compilatori GW-Basic (anche il Basic era interpretato) e presenta vantaggi e svantaggi. Lo svantaggio principale è una relativa lentezza rispetto ad un codice compilato veramente ed una scarsa post-ottimizzazione del codice, il vantaggio sta invece nel fatto che se si usano più programmi distinti, il modulo di RunTime può anche essere uno solo e si risparmia molto spazio sul disco.

I comandi per la compilazione sotto dBase IV si limitano pertanto al comando BDC (è scritto bene, sta per Borland dBase Compiler!) seguito dal nome del programma da compilare o dal nome del file che contiene la lista delle procedure e, opzionalmente, da una serie di switch che selezionano i parametri e le opzioni di compilazione.

L'operazione di compilazione è molto veloce ed assolutamente trasparente. Si deve solo fare attenzione al fatto che il compilatore legge sempre e comunque i file originali (.prg, .fmt ecc.) ignorando quelli già pre-elaborati dal compilatore interno al dBase.

I manuali sono due, il primo è lo stesso «Language Reference» fornito con il dBase IV, il secondo, specifico del compilatore, è veramente microscopico, sono 46 pagine comprese gli indici, e di queste ben più della metà sono relative alla potentissima Make, importata direttamente dal C++ della Borland che consente, in caso di modifiche fatte solo ad alcuni dei PRG che compongono l'applicazione, di ricompilare solamente quelle parti che sono state cambiate; risparmiando così tempo in fase di compilazione.

Cosa molto importante, i file compilati necessitano solo degli archivi e degli indici, tutti gli altri oggetti sono inclusi nel codice EXE e quindi lontano dalla portata dell'utente.

Anche il compilatore supporta il doppio Code-Page e include nel codice solo quella parte selezionata in fase di compilazione.

Col compilatore vengono anche forniti alcuni programmi di utilità che permettono, ad esempio, di dividere su più dischi un file e di ricomporlo in seguito o di vedere le prestazioni del Modo Protetto e dell'hardware in genere.

overlay chiamati più frequentemente in modo da ridurre la quantità di accessi alla rete. Tutta l'operazione di ottimizzazione e configurazione delle workstation di rete è molto semplice perché viene eseguita automaticamente da un apposito programma di installazione fornito insieme al dBase IV.

Nella versione 2.0 di dBase IV è stato aggiunto il supporto delle schede video VGA; se si utilizzano è possibile, in qualsiasi momento, settare il video in modo da avere 25, 43 o 50 righe e quindi una gran quantità di informazioni in più su un'unica videata.

Cosa c'è di nuovo

Nel riquadro sottostante sono rappresentati i principali passi dell'evoluzione del dBase IV dalla sua prima apparizione ad oggi. È evidente il salto tra il buon vecchio dBase Plus e la prima versione IV, ma anche le successive dimostrano un crescente affinamento delle prestazioni ed un evidente feedback col mercato che ha sollecitato ed ottenuto determinate prestazioni o funzionalità.

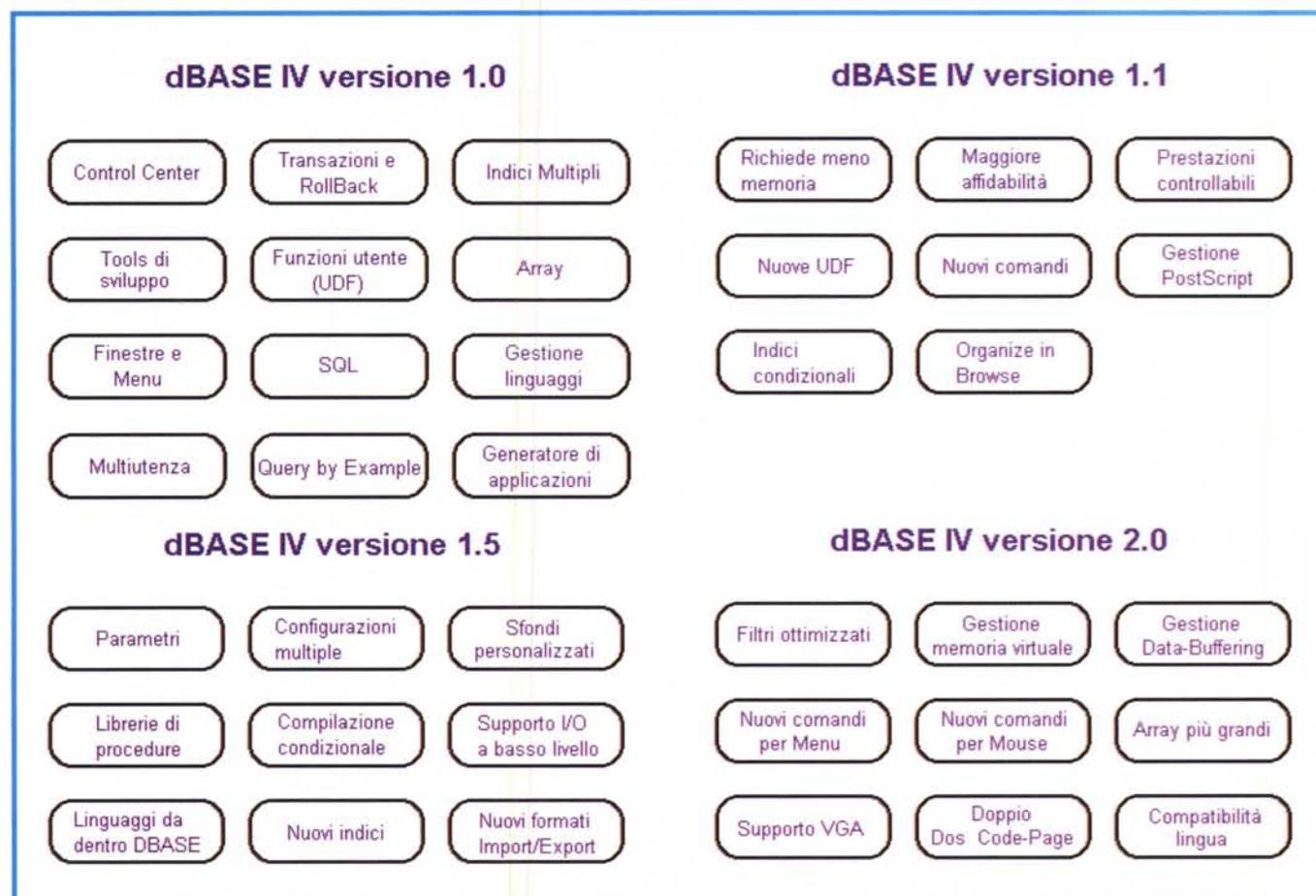
Tanto per citare le principali novità notiamo la gestione delle stampanti PostScript e degli indici condizionali nella 1.1 e poi le librerie, la compilazione condizionale e tutta una serie di funzioni orientate agli sviluppatori nella versione 1.5.

Con la versione 2.0, la prima interamente sviluppata in casa Borland, si nota l'attenzione rivolta contemporaneamente all'utente finale e agli sviluppatori professionisti. Per l'utente finale si è cercato di velocizzare le fasi di lavoro più frequenti, ad esempio molte ricerche sia sotto indice che con filtri, poi si è aggiunto il supporto delle schede VGA che consente di avere più informazioni in una sola finestra e il gestore autonomo di memoria virtuale che consente di lavorare anche con macchine che non dispongono di grandi risorse di RAM. Per ottenere gli aumenti di velocità desiderati la Borland ha dovuto riscrivere gran parte delle routine di ricerca utilizzando un proprio algoritmo intelligente denominato WarpSpeed; l'uso di questo algoritmo, e di un più oculato sfruttamento degli indici esistenti sia

permanenti che temporanei, ha consentito di migliorare anche di dieci volte le prestazioni delle ricerche e dello scrolling del Browse quando si utilizza un filtro.

Per gli sviluppatori o comunque per l'utente esperto sono stati aggiunti 12 nuovi comandi e potenziati altri 19 la maggior parte dei quali orientati all'uso del mouse e alla gestione degli eventi, ad esempio il fatto che il mouse venga posizionato o allontanato da un menu o da una sua voce (attenzione, solo posizionato senza alcun altro intervento) oppure le funzioni da attivare prima o dopo l'apertura di un menu o il riconoscimento di un click dentro una GET. È stata poi aumentata la dimensione degli elementi degli array che adesso possono arrivare a 65535, è stato reso ottimizzabile il data-buffering e, come già visto, ci sono due Code-Page (850 e 437) in modo da poter gestire correttamente anche le tastiere e i video nazionalizzati.

Con la versione 2.0 Borland ha anche mosso i primi passi per la trasportabilità del dBase sotto Windows infatti, al mo-





Ecco come si presenta una finestra di dBase IV sotto Windows. L'applicazione che si vede è quella creata apposta per la prova dal generatore di applicazioni di dBase IV.

La finestra di dialogo di un Query By Example. Il menu selezionato evidenzia le nuove possibilità di ottimizzazione del dBase IV. In questo caso gli indici di lavoro verranno anche conservati.



mento, è previsto solo il multitasking e la compatibilità dell'uso in modo protetto; ma si presume che, così come è già successo per Quattro e Paradox, anche il dBase IV finirà per migrare sotto ambiente Windows o OS2.

Il Control Center

Non è una novità del dBase IV versione 2.0, ma è sicuramente un ottimo motivo per passare dalla versione III alla IV, se ancora non si è fatto. Anche se personalmente sono convinto che un utente finale non troppo smaliziato si trovi meglio con Paradox, il Control Center è una gradita evoluzione del modo «Assist» del dBase III.

La filosofia è quella di avere sempre sott'occhio tutta la famiglia di oggetti relativi ad una certa applicazione. La base di partenza è un file di Catalogo, creabile dall'utente, che tiene traccia

degli oggetti legati funzionalmente ad un certo archivio e delle relazioni tra gli archivi e i rispettivi indici.

Una volta scelto un catalogo le sei sezioni in cui è divisa la finestra del Control Center vengono riempite ciascuna con gli oggetti relativi. La prima finestra contiene i Dati (gli archivi DBF), la seconda i Queries e le viste, la terza i Format, la quarta i Report, la quinta il Label ed infine la sesta le applicazioni.

Per quanto riguarda i dati, una delle grosse novità del dBase IV è la possibilità di creare dei file indice multipli (MDX) che possono contenere sino a 47 indici del tutto equivalenti a quelli NDX del dBase III e che sono legati al database dal fatto di avere stesso nome. Quindi mentre con il dBase III era difficilissimo, se non impossibile, sapere a quale archivio appartenesse un certo indice, adesso è molto facile aprire un file DBF e tutti gli indici che lo riguardano,

almeno quelli fondamentali. Sempre dal Control Center si possono anche reimpostare o modificare gli indici esistenti, se ne possono aggiungere di nuovi e si possono anche creare degli indici temporanei che il dBase IV cancella automaticamente appena terminata l'operazione.

Oltre ad aprire i database e gli indici tramite il Control Center si possono attivare anche le maschere di edit e di browse, preparare e stampare i report o le etichette, attivare le relazioni con altri database tramite i comandi di SQL ed infine si possono lanciare eventuali applicazioni (PRG) che compiono operazioni più complesse come ad esempio un ricalcolo da fare su tutto l'archivio.

Grazie a queste funzioni è ora possibile lavorare in dBase anche solo attraverso il Control Center quasi come da dentro un programma, si aprono i file e gli indici, si possono aggiungere nuovi record e si possono modificare o cancellare quelli esistenti, poi, quando serve, si passa a stampare i tabulati o le etichette. Per quanto riguarda le ricerche e la visualizzazione si accede al potentissimo SQL che permette di fare delle selezioni disegnando a video un esempio di quello che vogliamo ottenere. Dentro al Control Center ci sono anche le Macro, si attiva il modo apprendimento, si sceglie un tasto a cui associare la macro e si registrano le operazioni; una volta terminato si chiude la macro e, se servirà ancora, la si registra nella libreria del dBase IV. Tutte le macro sono editabili e possono anche contenere al loro interno delle altre macro.

L'ultima colonna del Control Center permette oltre al lancio di applicazioni dBase anche l'accesso al Generatore di Applicazioni (vedi riquadro), una delle più interessanti novità del dBase IV rispetto alla precedente versione III. Il generatore di applicazioni permette di creare in pochissimi minuti tutti i menu di una applicazione, definire la sequenza delle operazioni e attivare i principali controlli di flusso del programma che si sta generando.

In pratica la fase di lavoro tipica per chi vuol creare una applicazione in dBase IV consiste nel preparare dal Control Center tutti gli archivi, tutti gli indici, i Query, le maschere, i report e i formati vari che dovranno essere utilizzati all'interno del programma finale. Fatto questo si passa a scrivere eventuali routine (chiamate batch dal generatore di applicazioni) che devono eseguire i calcoli intermedi, i trasferimenti di dati o le elaborazioni più complesse. Una volta preparati gli oggetti singoli questi vengono assemblati e legati ai rispettivi menu dal generatore di applicazioni.

L'applicazione creata può girare sotto dBase con il classico comando Do <nome del programma> oppure può essere compilata e diventare un vero e proprio programma eseguibile in qualsiasi momento anche da chi non possiede il dBase IV.

Lo Structured Query Language

L'SQL è un linguaggio orientato alla consultazione e all'aggiornamento dei database. Nato inizialmente sui mainframe è stato poi adottato da molti svilup-

patori di database per PC perché consente, con pochi potenti comandi, di realizzare delle operazioni che richiederebbero decine di istruzioni dBase. Quello utilizzato da dBase IV è un sottinsieme dell'SQL DB2 utilizzato nei mainframe da IBM.

L'SQL è un linguaggio relazionale e quindi permette di vedere tutti i database aperti e attivare delle viste particolari che altro non sono che delle collezioni di campi tra archivi differenti legati da precise relazioni e che soddisfano determinate condizioni.

Una tavola di SQL contiene l'elenco dei database attivi, i campi che determinano le relazioni, l'elenco dei campi che determinano la vista finale e infine le condizioni di filtro. Ad esempio se abbiamo un archivio con i dati anagrafici dei clienti e un altro con gli ordini da questi fatti, e vogliono vedere il fatturato delle sedi di Milano cliente per cliente, basterà aprire da SQL i due archivi, metterli in relazione tramite il campo «codice cliente» (che deve esistere ovviamente in tutti e due gli archivi) e infine chiedere di vedere il campo «Nominativo» e di generare il campo «SOM-

Il generatore di applicazioni

Se per l'utente singolo è sufficiente il Control Center, eventualmente associato a qualche piccolo programmino di poche righe, per lo sviluppatore di software per conto terzi è invece indispensabile creare una interfaccia tra il dBase e l'utente, spesso inesperto, che deve contemporaneamente aiutare chi usa il programma nel lavoro da svolgere e proteggere dati e archivi da manipolazioni involontarie.

Il tipo di interfaccia ormai più diffuso è quello a menu e finestre, la cui realizzazione però è estremamente onerosa se si dovesse scrivere ogni singola riga di programma per generare i menu, stabilirne la grandezza, la posizione sul video, il colore del testo e dello sfondo, i titoli e i bordi. Poi si devono definire le singole voci di ciascun menu e le azioni ad esse associate, controllare la sequenza delle operazioni (ad esempio non si può scegliere «chiudi» se non c'è nulla aperto) e quindi l'abilitazione o meno di certe voci a seconda della fase di lavoro o dell'utente che sta utilizzando il programma (in rete il dBase IV può sapere chi è l'utente che sta lavorando).

Anche le singole maschere di inserimento, nel caso che l'applicazione debba essere assolutamente protetta, richiedono una grossa mole di lavoro necessaria a verificare che ciascun dato digitato sia compatibile con quanto richiesto. In questo compito lo sviluppatore è aiutato dalla possibilità che ha il dBase IV di verificare la validità di un inserimento prima che il dato sia accettato. Il controllo di validità si può ottenere sia obbligando la scelta del contenuto del campo ai dati di una tabella (e in questo caso l'utente può selezionare da un menu pop-up il dato prescelto) sia verificando che i valori immessi soddisfino certe condizioni. Il dBase IV permette anche di avere in una maschera o in un report dei campi calcolati, il cui valore cioè dipende dal contenuto degli altri campi presenti.

Tutte queste preparazioni si possono realizzare dal Control Center oppure dalla modalità punto. Una volta pronti tutti gli oggetti di una procedura si chiama il genera-

tore di applicazioni e si procede alla creazione dei menu ed alla aggregazione degli oggetti in una procedura.

La generazione di un programma inizia con l'assegnazione dell'ambiente di lavoro, cioè di tutta una serie di dati che identificano, ad esempio, il programmatore, i file di dBase che devono essere aperti prima di eseguire la procedura, della schermata iniziale di presentazione e altre cose simili.

Definita la parte introduttiva si può iniziare a creare il primo menu che, molto probabilmente, sarà la barra del menu principale. Subito dopo di devono creare i Pop-up menu che vengono attivati dal menu principale, ad esempio File, Edit, eccetera. Si deve sempre tener presente che il modo di procedere del generatore di applicazioni è quello di assemblare o collegare oggetti già esistenti. Quindi prima di poter dire che la prima voce del menu principale deve chiamare il menu «file» quest'ultimo deve già esistere.

Se si tiene ben presente questo modo di procedere la creazione di una applicazione è semplice e veloce. Se invece al momen-

to di assegnare un oggetto questo non è stato ancora creato o definito, si dovrà uscire dal generatore di applicazione, tornare al Control Center e creare l'oggetto mancante; non che la cosa sia impossibile, anzi, però lavorare in questo modo fa perdere un sacco di tempo e spesso può generare confusione in chi sta sviluppando l'applicazione.

Quindi, a patto di eseguire correttamente ed ordinatamente il lavoro di preparazione, il generatore di applicazioni è senza ombra di dubbio uno degli strumenti più interessanti a disposizione del programmatore che voglia utilizzare il dBase IV per sviluppare procedure per conto terzi.

Infine il generatore di applicazioni, grazie al fatto che è definibile in ogni situazione un help contestuale, riesce a creare tutta la documentazione tecnica relativa assemblando le immagini delle maschere, dei menu e degli help scritti dallo sviluppatore. Il file generato può essere ripreso con un qualsiasi Word Processor per essere trasformato, con pochi ritocchi, in un perfetto manuale utente.

Una delle prime fasi di lavoro mentre stiamo creando un'applicazione di prova. Si tratta di un semplice gestore per l'indice di Mcmicrocomputer, per farlo non ci sono voluti più di 20 minuti.



Il Debugger

Sviluppare un'applicazione che non sia la semplice rubrica telefonica senza avere accesso ad un debugger è una impresa difficile per chiunque. Ben lo sanno gli sviluppatori che utilizzavano il dBase III che, sebbene fosse un linguaggio interpretato, erano costretti a piazzare dei Print qua e là in giro per i programmi al solo fine di vedere che cosa stesse realmente succedendo.

Finalmente con il dBase IV il Debugger viene fornito integrato al pacchetto e

consente di posizionare dei break nei punti strategici, di osservare in tempo reale il valore delle variabili che si desidera tenere sott'occhio, di vedere il nome dei file aperti e i relativi indici e, come se questo non bastasse, di veder scorrere le righe del sorgente man mano che vengono eseguite (volendo anche in modalità passo passo).

Si può commutare in qualsiasi momento tra la finestra di output dell'applicazione e quella di Debug e, in caso di errore, quest'ultima viene richiamata automaticamente.

Per attivare il Debugger si può utilizzare sia l'apposito menu del Control Center sia attivare il set Trap direttamente dal modo punto.

Il Debugger purtroppo non funziona con i programmi .EXE fatti dal compilatore.

```

C:\DBASE\EXAMPLES\COMP\GOODS.PRG
182 * Turn off filter if no matching records found
183 DO Warnoff
184 DO Show_msg WITH "No Goods (inventory) records match the " +
185 "filter condition"
186 SET FILTER TO
187 GO record_num
188 ENDIF
189 ENDIF
110 ELSE
111 IF choice = "T"
112 * If user selects "T", turn off filter

```

DISPLAY		BREAKPOINTS	
List_flds	: PART_ID, PART_NAME, QT	1:	
Key2	: NONE	2:	
Look_txt	: Variable not found: LO	3:	
		4:	

```

DEBBUGER
Mark Area: 1 Database file: Program file: goods.prg
Record: 8 Master Index: Procedure: GOODS
ACTION: Current line: 45
Stopped for step

```

MA di Totale Fattura» (che non esiste in nessuno dei due database) per tutti i clienti con il codice della sede corrispondente a quello di Milano. Tutta l'operazione viene effettuata in una maschera video e non si devono scrivere nemmeno due righe di codice. La stessa maschera può però essere trasformata (e lo fa dBase automaticamente tramite il generatore di applicazioni) in una serie di comandi SQL da inserire in una procedura dBase. Anche il compilatore è in grado di includere nel programma finale tutte le tavole di SQL utilizzate nell'applicazione.

Per velocizzare le ricerche si possono creare fino a 47 degli indici temporanei per ciascuna maschera di SQL; questi indici vengono automaticamente gestiti da SQL che si preoccuperà di aprirli e di mantenerli aggiornati finché si utilizza la tavola di SQL.

Due sono le modalità operative di SQL, la prima è il Qbe (Query by example) praticamente quella vista nell'esempio precedente, in cui il risultato è una vista su parte dei record che soddisfano certe condizioni, la seconda modalità è il Qbu (Query by update) il cui risultato è una modifica di quei dati che soddisfano le condizioni previste, ad esempio la modifica del prefisso telefonico di tutti i clienti di Londra il cui numero inizia per 441.

Ma la potenza di un Query non si esaurisce qui se si considera che sul manuale del dBase IV ci sono 120 pagine dedicate solo all'uso di SQL.

L'Help e la documentazione

Il dBase IV viene fornito con una poderosa dotazione di manuali ovviamente in inglese. Si comincia con quello per l'installazione e la configurazione «Getting Started» che contiene anche una panoramica sulle novità della versione 2 e una introduzione al dBase. Finite le 280 pagine di «Getting Started» si possono affrontare le quasi 500 di «Using dBase IV» destinato all'uso interattivo del dBase, quindi alla creazione dei file, all'uso degli strumenti di disegno, del Control Center e del generatore di applicazioni.

L'utente medio si può fermare qui, ma il programmatore deve necessariamente passare al «Programming in dBase IV», 530 pagine dove viene spiegato il linguaggio, l'SQL e come ottimizzare le prestazioni o modificare i Template che sarebbero poi i programmi che permettono di disegnare i Form, i Report e anche il Control Center. Finito di leggere il «Programming in dBase IV» si può finalmente iniziare a scrivere qualche programma tenendo però ben a portata di mano il massiccio «Langua-

ge Reference» le cui 790 pagine equivalgono in pratica alla Bibbia del dBase IV.

Per fortuna tutte le fasi di lavoro del dBase sono assistite da un efficientissimo Help contestuale, quindi in caso di dubbio, la pressione del tasto F1 dovrebbe risolvere i dilemmi più immediati. Unico appunto da fare alla manualistica, e in particolare alla Bibbia, è la mancanza di evidenziazione delle modifiche apportate nelle varie versioni (sarebbe bastato un asterisco vicino al nome) e la scarsa attenzione alle problematiche di condivisione dei file in rete. Per fare un esempio non è chiaro in quali casi sia necessario l'uso esclusivo per la creazione di indice temporaneo di SQL, e questo per un manuale destinato agli sviluppatori è piuttosto grave. Insomma ci sarebbe piaciuto di più se avesse avuto un capitolo dedicato all'uso in rete o sul supporto DPMI sotto Windows e OS2.

Conclusioni

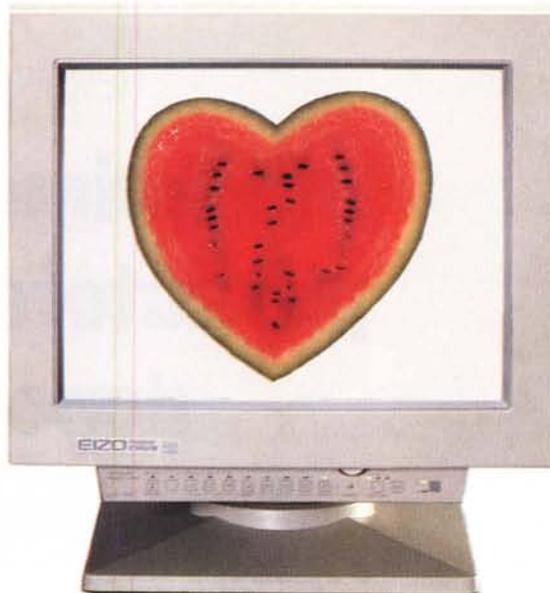
La valutazione del dBase IV e del relativo compilatore dipendono molto dal tipo di uso che si prende in considerazione. Il dBase IV è infatti contemporaneamente un pacchetto per l'utente finale e un potente linguaggio per lo sviluppatore. Se lo si guarda dal punto di vista del database fine a se stesso non regge assolutamente al confronto con pacchetti molto più potenti e versatili, come ad esempio il Paradox per Windows, gli mancano, tanto per citarne due, la gestione delle immagini e qualsiasi possibilità di interagire con altri database che non siano in formato DBF.

Se invece lo si considera come un linguaggio le cose cambiano completamente, la potenza e la semplicità dei suoi comandi, il generatore di applicazioni, l'SQL e soprattutto il compilatore compatibile al 100% con il codice interpretato, gli danno un ampio margine di vantaggio nei confronti di altri linguaggi simili ma che non sono ormai compatibili che in minima parte. Certo anche qui si sente un po' la mancanza di un tool grafico, ma per un programmatore esperto non è affatto difficile interfacciarsi con una delle tante librerie che già esistono.

E se il tempo è denaro, e lo è soprattutto per una software house, la rapidità con cui in dBase IV si riesce a sviluppare un prodotto rifinito e funzionale è assolutamente impagabile. Se poi si realizzerà presto la prevista migrazione in ambiente Windows allora il successo è garantito.

MB

AMORE A PRIMA VISTA.



Di un monitor EIZO non occorre dire molte cose: basta guardarlo, guardarlo e guardarlo ancora, per accorgersi che davvero gli occhi non si stancano mai. Quindi, se avete finalmente deciso che il vostro lavoro e i vostri occhi meritano strumenti migliori, fate pure i vostri confronti. E vinca il migliore.

Confronto monitor 17 pollici a schermo piatto ad alte prestazioni	EIZO F5501.W		
Visione a tutto schermo	si		
Dot pitch	0,28		
Controllo dinamico fuoco/pixel	si/si		
Risoluzione massima n.i.	1280 x 1024		
Freq. scansione verticale a 1024 x 768	75 Hz		
Trattamento antiriflesso/antistatico	Ergo Coat		
Controllo del colore	si		
Economizzatore di energia	Power Manager		
Norme di protezione antiradiazioni	MPR II		
Protezione anti interferenza	si		
Prezzo (IVA esclusa)	2.390.000		
Impressione visiva dal vero	OTTIMA!		

EIZO[®]
Professional Display Systems

Strumenti per vedere meglio.

I monitor EIZO sono distribuiti da Epson Italia S.p.A. - 20099 Sesto S. Giovanni (Milano) - V.le F.lli Casiraghi, 427 - Fax 02/2440750

Se desideri saperne di più
inviaci il coupon, o mandaci un fax,
o chiama il numero verde



Si, vorrei maggiori informazioni:

sulla gamma dei vostri monitor sul sistema di Power management sull'Ergo Coat

Nome e Cognome/Società _____
via _____ C.A.P. _____

Ln tutto il mondo

Greenwich è il riferimento unico per misurare il tempo. Ma da noi c'è qualcos'altro. Un riferimento per conoscere e scegliere gli strumenti che lo misurano: Orologi. Una rivista pensata con passione; nelle sue pagine il mondo del tempo, in tutte le sue forme: tecnica, storia, curiosità e futuro. Splendide immagini di orologi moderni e antichi accompagnano informazioni precise e articoli attenti e puntuali sulla tecnica, la cultura del tempo e sulle rarità. Insomma una guida sicura che non ha paralleli: proprio come il meridiano di Greenwich.

Il riferimento più autorevole dopo il meridiano di Greenwich.

technimedia

Pagina dopo pagina, le nostre passioni.

Orologi®

LE MISURE DEL TEMPO

technimedia

Ulysse Nardin:
la tecnica del Tellurium

I segreti di
Alain Silberstein

Orologi. I primi sui secondi.