

Funzionalità trasversali in Windows 3

di Francesco Petroni

Windows 3, l'ambiente Windows 3 nel suo complesso, costituito dal prodotto Windows 3 e dall'insieme dei prodotti che vi girano, dispone di una serie di funzionalità uguali, utilizzabili in tutti gli applicativi e attivabili in ciascuno di questi, con modalità relativamente simili, anche dai normali utenti. Dedichiamo a queste funzionalità, che battezeremo per l'occasione «Funzionalità Trasversali», una serie di articoli specifici, in cui, oltre ad esporre il significato della funzionalità stessa, cercheremo anche di realizzare un'applicazione uguale, con le necessarie varianti, con più prodotti differenti

Utilizzeremo applicativi finali evoluti (in questo articolo Excel, WinWord, ToolBook, Object Vision e Visual Basic) escludendo quindi i linguaggi, sia perché non ci competono come argomento, sia perché i destinatari di questa rubrica Windows dovrebbero essere tutti gli utenti Windows e non solo i programmatori, che ovviamente preferiscono utilizzare, data la loro specializzazione, i vari C, C++, Pascal e le varie librerie di funzioni Windows.

Abbiamo cominciato nel numero scorso con la trattazione del DDE, tema affascinante ed in continua evoluzione sul quale torneremo più volte, e continuiamo in questo affrontando il tema Dialog Box (in italiano tradotto in Finestra di Dialogo, ma diremo più frequentemente Dialog Box). Proseguiremo nei prossimi numeri affrontando l'argomento menu e l'interessantissimo, e ancora per molti aspetti «misterioso», argomento DLL.

A cosa serve e come funziona una Dialog Box

La Dialog Box è una delle numerose modalità attraverso le quali avviene il colloquio tra l'utente e l'applicazione Windows. Le altre modalità sono il menu, anche questo ormai oggetto standard presente e funzionante sempre allo stesso modo in qualsiasi applicazione, e le più semplici Message Box, che servono per l'Input di una singola risposta necessaria all'applicazione per andare avanti. Vediamo due Dialog Box in figura 1 e 2.

Una ulteriore modalità di colloquio è costituita dal Bottone, che può essere quasi sempre (o meglio, in quasi tutti i prodotti) iconizzato, ovvero associato ad una piccola

figurina esplicativa, e che serve per attivare una determinata azione. Il Bottone in genere risiede in una Dialog Box e quindi la sua trattazione dovrebbe ricadere in quella della Dialog Box stessa. In alcuni casi però i Bottoni possono essere inseriti anche al di fuori. È il caso ad esempio dell'Excel 3 e del WinWord che permettono di inserirne rispettivamente in un foglio di calcolo o in un documento, e quindi al di fuori sia dei menu sia delle Dialog Box sia delle Message Box.

Torniamo alla Dialog Box. È sempre costituita da una Cornice che racchiude una serie di Oggetti Attivi, su cui può intervenire l'operatore, e da una serie di Testi Fissi, in genere esplicativi del significato dei vari oggetti. In tal modo la Dialog Box diventa un elemento autoesplicativo, semplice da capire e semplice da utilizzare anche dall'utente estemporaneo.

Oggetti Attivi sono ad esempio una TextBox, in cui l'operatore digita una Stringa, una ListBox, in cui l'operatore sceglie uno dei valori predefiniti che scorrono nella Box, un CheckBox, che può essere acceso o spento con un Click, infine un Bottone che può essere pigiato per lanciare l'esecuzione di un sottoprogramma.

Se si sta programmando una qualsiasi Dialog Box, ad ogni suo oggetto andrà in genere associata una variabile. A questa dovrà essere assegnato, prima dell'attivazione della Dialog Box stessa, un valore iniziale. In uscita dalla Dialog Box la variabile assumerà il suo valore finale, che dovrà essere usato in un momento successivo dal programma.

Il tipo di variabile dipende ovviamente dal tipo di Oggetto. Quella associata alla TextBox assume un valore Stringa o Numero, quella associata alla ListBox assume in generale un valore numerico, che indica la posizione che la voce scelta ha nell'elenco delle voci. La variabile del CheckBox è un valore logico mentre quello assunto dal Bottone è un valore numerico e/o logico.

Il funzionamento della Dialog Box, indipendentemente dalla sua complessità e dalla sua organizzazione interna, comporta sempre tre momenti logici successivi.

Il primo è costituito dalla impostazione iniziale assunta dalle variabili. Queste saranno azzerate se la Dialog Box, serve per una nuova immissione, oppure assegnate a valori preesistenti se questi debbono essere modificati. Le modalità di inizializzazione delle variabili non dipendono dalla Dialog Box, ma dal prodotto che si sta usando.

Il secondo momento è il Lancio della Dialog Box, sulla quale l'operatore agisce con

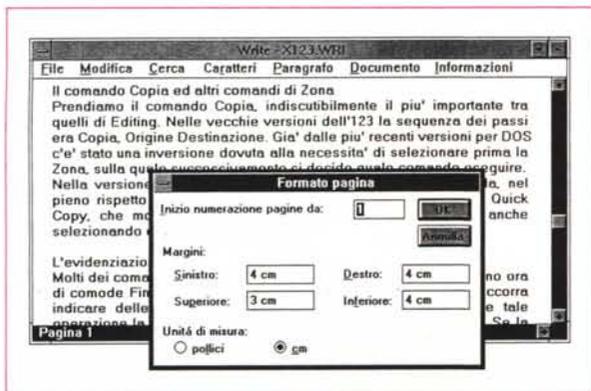
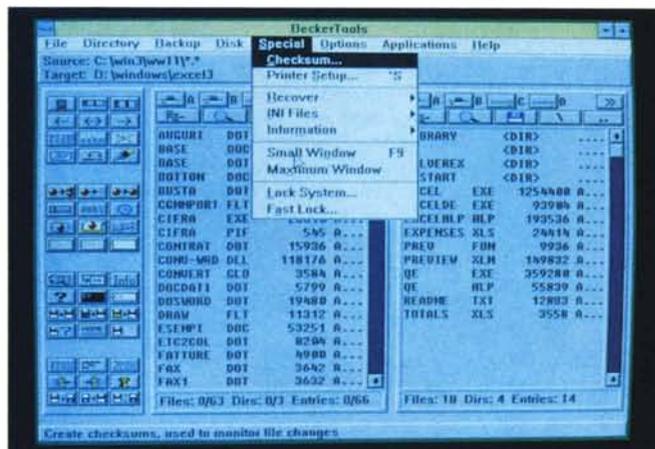


Figura 1 - Dialog Box in Windows 3 - Una normale Dialog Box operativa.

Quando l'utente lavora con Windows, o con uno qualsiasi dei suoi programmi, il dialogo con l'applicazione avviene prevalentemente attraverso la Dialog Box (in italiano: Finestra di Dialogo), strumento operativo messo a disposizione dall'ambiente e richiamabile in svariati modi non solo dai vari programmi Windows, ma anche dall'utente stesso che trova funzionalità di generazione di Dialog Box in ogni applicativo evoluto.

Figura 2 - Dialog Box in Windows 3 - Esempio di una Box ricca. Qui vediamo una Dialog Box particolarmente complessa (si tratta delle Becker Tools, una serie di utility tipiche di un File Manager) in cui sono concentrati tutti i numerosissimi comandi operativi dell'applicazione. L'«oggettistica» interna ad una qualsiasi Dialog Box è sempre riferibile ad una dozzina di tipi, comuni a tutte le Dialog Box di tutti i prodotti.



le classiche modalità Windows. I tasti Tab per spostarsi in sequenza tra i campi, oppure i tasti Alt+carattere se ai vari campi sono state assegnate delle scorciatoie via tastiera. Invio per uscire con OK, Esc per uscire con Annulla.

Il terzo momento è proprio l'Uscita. La modalità di uscita può variare da prodotto a prodotto. L'uscita standard si ottiene ricorrendo ai soli due Bottoni, OK e Annulla, citati prima e il cui significato è standard. OK conferma tutte le impostazioni inserite o Annulla le annulla, appunto.

Le modalità di uscita variano però da prodotto a prodotto. Con l'Excel 3 ad esempio è possibile inserire più Bottoni (come abbiamo fatto noi) per cui in uscita è possibile percorrere più routines differenti, dipendenti dal Bottone che è stato premuto.

WinWord invece dispone solo dei due Bottoni minimi (OK e Annulla), per cui le routine conseguenti possono essere solo due.

In prodotti più evoluti esistono altre modalità di uscita, ad esempio al verificarsi di un dato evento, non necessariamente legato ad un Bottone per l'uscita.

Questo discorso però ne comporta un altro più generale che si può ricondurre alla semplice domanda: cosa avviene all'interno della finestra di Dialogo durante la sua digitazione?

Ad esempio se nella finestra, come nel caso che proporremo, è previsto un calcolo riguardante due degli elementi della finestra stessa, questo calcolo quando verrà eseguito?

Diciamo subito che nei prodotti meno programmabili il calcolo avviene solo in fase di uscita per cui il suo risultato non si potrà vedere subito nella Dialog Box. Nei prodotti programmabili invece si può programmare quando e come, anche durante l'uso della Dialog Box, far avvenire il sospirato calcolo.

Questo è un elemento di differenziazione che deve essere ben noto all'utilizzatore che da una parte non può pretendere dal prodotto che usa cose che lo stesso non è in grado di fare, e dall'altra deve invece sfruttare al meglio le potenzialità del prodotto che sta usando.

Il nostro «CASE STUDY» e gli oggetti necessari alla sua soluzione

Dopo questa introduzione teorica passiamo alla pratica descrivendo nel dettaglio il nostro caso studio, che proponiamo per

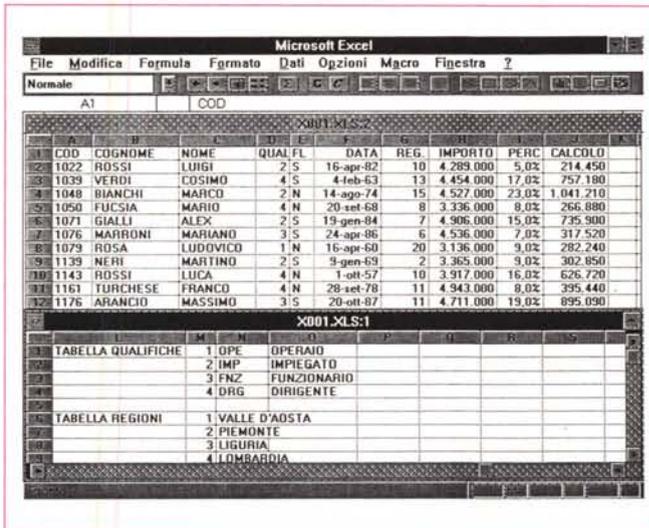
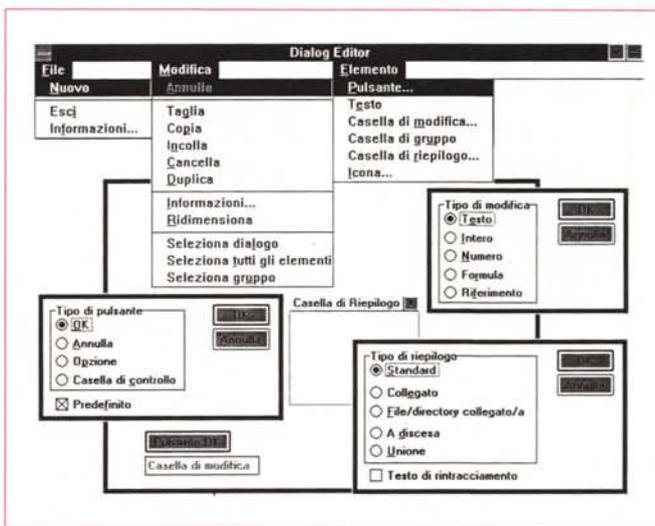


Figura 4 - Dialog Box in Windows 3 - Il Dialog Box Editor. L'editore di Dialog Box, inserito nella confezione di Excel 3, rende la produzione della Box un'operazione esclusivamente grafica da svolgere con gli appositi Tools su una finestra inizialmente vuota. Il risultato dell'operazione è la codifica standard riconosciuta sia dall'Excel che dal Winword, che una volta inserita in una Macro viene convertita in una Box a video.

Figura 3 - Dialog Box in Windows 3 - Il nostro archivio in Excel 3.

Lo scopo che ci prefiguriamo nell'articolo sulle Dialog Box è quello di realizzare una stessa Box, destinata all'inserimento di una serie di dati, con più prodotti. È chiaro che, a seconda del prodotto, l'uso che faremo dei dati inseriti nella Dialog Box varierà. Con l'Excel i dati saranno scaricati nel foglio, con il WinWord saranno scaricati in un documento, con il Visual Basic in un file sequenziale, e infine con l'Object Vision e il Tool-Book, in un file DBF.



chiarezza in Excel, e che comporterà l'utilizzazione di una larga tipologia di oggetti standard presenti nelle Dialog Box.

In figura 3 vediamo un foglio Excel in cui i dati che vogliamo gestire con la Dialog Box sono posti in forma tabellare. Alcuni di questi dati sono dei codici che vanno ovviamente «scodificati» secondo una tabella di corrispondenza che si può vedere nella parte inferiore della figura. Delegeremo alle varie Dialog Box anche questo compito.

Una volta chiarite le varie problematiche inserite del «caso studio» passeremo alla realizzazione, con i vari prodotti in esame, della stessa Dialog Box, che deve servire per acquisire dei dati, di vario tipo, che andranno a finire... da qualche parte.

La sua realizzazione avviene a seconda dei prodotti con due modalità ben differenti. In alcuni occorre scrivere le specifiche di ciascun oggetto in un programma, in altri si lavora in maniera assolutamente grafica disegnando gli oggetti e posizionandoli con il mouse.

Con il prodotto Excel viene anche «regalato» il Dialog Box Editor che serve proprio per «disegnare» la Dialog Box ed evitare in questo modo di dover scrivere a mano le codifiche nel foglio Excel (fig. 4). In uscita dal Dialog Box Editor il Clipboard di Windows contiene proprio tali codifiche direttamente incollabili nel foglio.

Una volta disegnata la Dialog Box, occorre decidere cosa fare dei dati gestiti dalla Dialog Box stessa.

Nel caso di Excel i dati acquisiti sono destinati a finire nel foglio di calcolo. Vanno invece nel documento se lavoriamo con WinWord, in un file Sequenziale o in un file Random, se lavoriamo con Visual Basic, in un file DBF, Paradox o in un canale DDE, se lavoriamo con Object Vision.

Con l'ultimo prodotto in esame, l'Asymetrix ToolBook 1.5, invece sfrutteremo la sua dotazione di librerie DLL (fortunatamente ben documentate), ed in partico-

Campo oppure Elemento	Cod.Num. in Excel 3	Istruzione WinWord	Tipo di Elemento nella Dialog Box	Commento riferito all'esempio
Finestra di Dialogo	(vuoto)	DialogBox	Finestra di Dialogo (definisce la cornice)	La sintassi prevede quattro valori di posizionamento: X, Y, L, H
Testo Normale qualsiasi	5	Text	Testo	Tutte le scritte fisse che appaiono nella Dialog Box
Codice	6	TextBox	Casella Testo	Input in formato stringa del Codice
Nominativo	6	TextBox	Casella Testo	Input in formato stringa del Cognome e Nome
Data di Inserimento	6	TextBox	Casella Testo	Input in formato stringa della Data
Dato Registrato S/N	13	CheckBox	Casella di Selezione	La casella assume i due valori Vero e Falso
Qualifica	11	OptionBox	Gruppo Pulsanti	La struttura prevede una cornice e al suo interno più Bottoni di Opzione
Tipi di qualifica	12	OptionButton	Pulsante Opzione	Bottoni di Opzione mutuamente esclusivi
Regione	15	ListBox	Casella di Riepilogo	Occorre far riferimento ad una Lista esterna alla zona di definizione. Il risultato restituito è un numero.
Importo	8	non disponibile	Casella Numero	Il valore accettato è un numero
Percentuale	8	non disponibile	Casella Numero	Il valore accettato è un numero
Calcolo	9	non disponibile	Casella Formula	Il valore accettato è una formula
Calcola	1	OkButton	Pulsante OK predefinito	In WinWord sono possibili solo i due bottoni standard OK e Annulla
Registra	1	OkButton	Pulsante OK predefinito	In Excel 3 sono possibili più bottoni tipo 1. In uscita la funzione Finestra di Dialogo assume valori differenti
Cancella	1	OkButton	Pulsante OK predefinito	
Interrompi	1	OkButton	Pulsante OK predefinito	
Annulla	2	CancelButton	Pulsante Annulla predef.	
Altre Tipologie	vari altri	pochi altri		In Excel 3 esistono anche oggetti di File Manager

Figura 5 - Dialog Box in Windows 3 - Tipologia degli oggetti inseribili in una Dialog Box. L'esposizione teorica sulle tipologie di oggetti inseribili in una Dialog Box la faremo utilizzando Excel 3, sia perché è il prodotto più diffuso tra quelli presi in esame, sia perché è quello in cui l'oggettistica Dialog Box è rigorosamente codificata ed è ben documentata nel manuale, e WinWord 1.1, in cui le Dialog Box non sono invece assolutamente documentate. Nella tabella, nostra rielaborazione dal manuale Excel, vediamo codice dell'oggetto, suo significato e valore restituito.

lare quella che dispone di una trentina di funzioni corrispondenti ad altrettanti comandi dBase III. In pratica ci proponiamo di attivare e dialogare direttamente con un archivio DBF.

In figura 5 abbiamo schematizzato la struttura del nostro archivio (che si può anche controllare nella precedente fig. 3), mettendola in corrispondenza con i vari elementi usati nella Dialog Box. Abbiamo anche inserito la corrispondente sintassi dell'elemento sia in Excel che in WinWord, più alcune note.

Questa tabella ci libera dalla necessità di dover descrivere nel dettaglio tutti gli elementi. In sede di conclusioni paragoneremo le implicazioni legate all'uso dei vari oggetti con i vari prodotti.

La Dialog Box sviluppata con Excel 3

Occorre realizzare una Macro che contiene una serie di istruzioni divisibili in due gruppi logici. Ne vediamo un significativo stralcio in figura 6, mentre in figura 7 vediamo la Dialog Box in azione.

Il primo gruppo contiene quelle istruzioni che vanno eseguite prima del lancio della Dialog Box, e che consisteranno essenzialmente nella attribuzione del valore iniziale voluto per ciascuno dei Campi, valore che va posto nelle varie celle della colonna Inizio/Risultato della Finestra di Dialogo.

Se tali celle sono state nominate (per farlo si usa l'istruzione Formula Definisci Nome),

	A	B	C	D	E	F	G	H	I	J	K
1	Inserimento dati con Finestra di Dialogo	Typo	X	Y	L	H	Testo	Inizio/Risultato	Valle d'Aosta		
2	=ATTIVA.FINESTRA("X001.xls")				480	280					Piemonte
3	=CELLA.VALORE(codice;"0000")	5	20	10			Finestra per Acquisizione DA				Liguria
4	=CELLA.VALORE(nome;"")	5	20	40			Codice				Lombardia
5	... omissis	6	120	38	60			100			Trentino Alto Adige
6		5	20	60			Nominativo				Friuli Venezia Giulia
7	=FINESTRA.DI.DIALOGO(%C2;%I28)	6	120	58	180			ROSSI			Veneto
8	=SE(%A7=23;VALA(%A30))	5	20	80			Data di Inserimento				Emilia Romagna
9	=SE(%A7=24;VALA(%A40))	6	180	78	120			11-apr-1991			Toscana
10	=SE(%A7=25;VALA(%A50))	13	20	110			Dato Registrato S/N	VERO			Marche
11	=FORMULA(MAIUSC(nome);"RC1")	14	320	40	140	90	Qualifica	2			Umbria
12	=FORMULA(codice;"RC2")	11						1			Lazio
13	... omissis	12		56			Operalo				Abruzzo
14	=FORMULA(INDICE(K1:K20;Regione);"RC7")	12		74			Impiegato				Molise
15	=RITORNO()	12		92			Funzionario				Campania
16		12		110			Dirigente				Puglia
17		5	280	150			Regione				Basilicata
18		15	280	170	180	100	R1C11:R20C11	2			Calabria
19		5	20	140			Importo				Sicilia
20		8	140	138	120			5000			Sardegna
21		5	20	160			Percentuale				
22		8	140	158	120			50			
23		5	20	180			Calcolo				
24		9	140	178	120			=250000			
25		1	20	220	90	18	Calcola				
26		1	20	250	90	18	Registra				
27		1	140	220	90	18	Cancella				
28		1	140	250	90	18	Interrompi				

Figura 6 - Dialog Box in Windows 3 La versione Excel 3.

Questa tabella comprende sette colonne successive: tipo dell'oggetto, sue coordinate X,Y e sue dimensioni L,H eventuale contenuto testuale. Importantissima è la settima colonna: Inizio/Risultato dove occorre inserire i valori iniziali che si vuole appaiano all'apertura della Dialog Box e dalla quale si possono poi prelevare i valori risultanti in uscita.

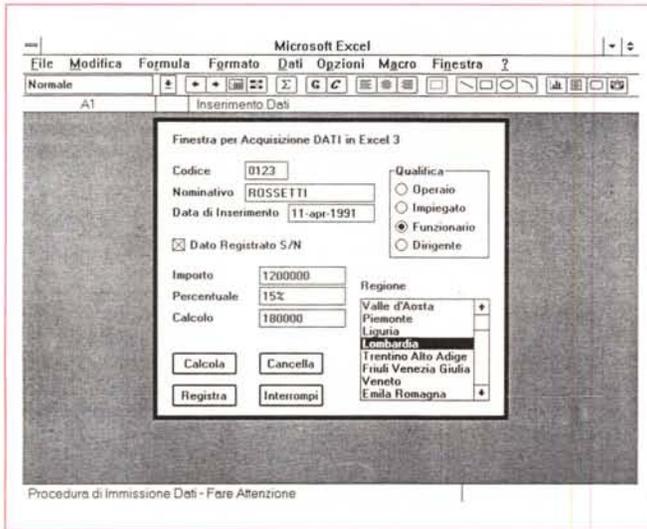


Figura 7 - Dialog Box in Windows 3 - Organizzazione della Macro Excel 3 che gestisce la Dialog Box.

In una procedura complessa la Dialog Box non esaurisce le varie necessità operative connesse alla gestione dei dati in essa presenti. Il richiamo di una Dialog Box può essere assimilato al richiamo di una subroutine, che permette di intervenire sui dati. In caso di inserimento di un nuovo record andranno preventivamente azzerati i dati iniziali, in caso di modifica di record esistente, andranno semplicemente utilizzati i dati propri di quel record, ecc.

cessive alla Finestra.di.Dialogo, contiene una serie di sottoprogrammi, uno per ciascun tipo di Uscita, e quindi come minimo ne contiene due in caso di presenza dei soli due Bottoni OK e Annulla.

L'istruzione «uguale e contraria» alla CELLA.VALORE, che serve quindi per trasferire il valore assunto da ciascuna delle celle della colonna Inizio/Risultato nella cella di destinazione (in genere su un foglio differente da quello Macro) è del tipo:

=FORMULA(Importo;R3C4)

dove è evidente che Importo assume il valore digitato nella Finestra di Dialogo e che R3C4 è il riferimento della cella di destinazione che può essere espresso in una

l'aspetto delle istruzioni di assegnazione sarà:

=CELLA.VALORE(Importo;100000)

che sta a significare che alla cella nominata Importo, viene attribuito il valore 100000.

I Campi di tipo CheckBox, OptionGroup, ListBox contengono dati codificati, per cui anche il loro valore di inizializzazione andrà codificato. Ad esempio se si volesse definire la «LOMBARDIA» come valore di default per la ListBox delle Regioni, occorrerà impostare il valore numerico 3 nella corrispondente cella della colonna Inizio/Risultato.

Dopo le istruzioni di inizializzazione, che dovrebbero essere all'incirca una per ogni Campo, si può finalmente inserire la istruzione Finestra.di.Dialogo <rif.> che necessita di un parametro costituito dal riferimento della zona del foglio in cui è stata inserita la codifica della Finestra di Dialogo stessa (sette colonne per x righe).

Tale tabella si sviluppa come detto ora ora su sette colonne ed è fortunatamente ben documentata nel manuale di Excel. In sintesi:

- col.1 Tipo dell'elemento
- col.2 Coordinata X
- col.3 Coordinata Y
- col.4 Larghezza
- col.5 Altezza
- col.6 Testo (eventuale, in quanto dipende dal tipo)
- col.7 Inizio/Risultato (eventuale, in quanto dipende dal tipo).

Si esce dalla Dialog Box premendo uno dei Bottoni di uscita, che sono quelli identificati dal codice numerico che va da 1 a 4. Nel caso però esistano più Bottoni con lo stesso codice, Excel, molto opportunamente, restituisce un valore numerico pari alla riga della tabella di definizione della Finestra di Dialogo in cui c'è il Bottone attraverso il quale si è verificata l'uscita. E tale valore viene assunto proprio dalla cella dove è stata messa l'istruzione Finestra.di.Dialogo.

Il secondo gruppo di istruzioni, quelle suc-

```

Sub MAIN
Dim AS(19), BS(4) : RS = Chr$(13) + Chr$(10)
AS(0)="Valle d'Aosta" : AS(1)="Piemonte" : AS(2)="Liguria" : AS(3)="Lombardia"
AS(4)="Trentino Alto Adige" : AS(5)="Friuli Venezia Giulia" : AS(6)="Veneto"
AS(7)="Emilia Romagna" : AS(8)="Toscana" : AS(9)="Marche" : AS(10)="Umbria"
AS(11)="Lazio" : AS(12)="Abruzzo" : AS(13)="Molise" : AS(14)="Campania"
AS(15)="Basilicata" : AS(16)="Puglie" : AS(17)="Calabria"
AS(18)="Sicilia" : AS(19)="Sardegna"
BS(0)="Operaio" : BS(1)="Impiegato" : BS(2)="Funzionario" : BS(3)="Dirigente"
Begin Dialog UtenteDialogo 220, 50, 400, 220
Text 30, 2, 360, 16, "Dialog Box per l'Acquisizione di Dati in MS WinWord"
Text 10, 30, 80, 16, "Codice"
TextBox 90, 27, 50, 18, .cd
Text 10, 50, 80, 16, "Nominativo"
TextBox 90, 47, 130, 18, .nm
Text 10, 70, 140, 16, "Data Inserimento"
TextBox 130, 67, 90, 18, .di
CheckBox 10, 100, 180, 16, "Dato Registrato S/N", .cb
GroupBox 240, 20, 150, 80, "Qualifica"
OptionGroup .qu
OptionButton 250, 35, 100, 16, BS(0)
OptionButton 250, 50, 100, 16, BS(1)
OptionButton 250, 65, 100, 16, BS(2)
OptionButton 250, 80, 100, 16, BS(3)
Text 10, 130, 80, 16, "Importo"
TextBox 100, 127, 120, 18, .im
Text 10, 150, 80, 16, "Percentuale"
TextBox 100, 147, 120, 18, .pc
Text 10, 170, 80, 16, "Calcolo"
TextBox 100, 167, 120, 18, .ca
Text 240, 105, 150, 18, "Regione"
ListBox 240, 120, 150, 100, AS(), .rs
OKButton 10, 190, 70, 16
CancelButton 100, 190, 70, 16
End Dialog
Dim ProvaDB As Dialog UtenteDialogo
Dialog ProvaDB

REM uscita brutale
Inserisci ProvaDB.cd + r$
Inserisci UCCase$(ProvaDB.nm) + r$
Inserisci ProvaDB.di + r$
If ProvaDB.cb = 1 Then
Else
Inserisci "Dato Registrato" + r$
Else
Inserisci "Dato non Registrato" + r$
End If
Inserisci AS$(ProvaDB.rs) + r$
Inserisci BS$(ProvaDB.qu) + r$
Inserisci ProvaDB.im + r$
Inserisci ProvaDB.pc + r$
Inserisci Str$(Val(ProvaDB.im) * Val(ProvaDB.pc) / 100) + r$

REM uscita posizionata - tipo 1 - con segnalibro
ModificaVaIA "aa"
ParolaDestra 1
Inserisci r$ + UCCase$(ProvaDB.nm)
ModificaVaIA "bb"
ParolaDestra 1
Inserisci r$ + ProvaDB.di
' .. omississ

REM uscita posizionata - tipo 2 - con campo
' .. omississ

End Sub

```

Figura 8 - Dialog Box in Windows 3 - La Macro WinWord che attiva la Dialog Box.

Le stesse specifiche che in Excel vanno poste in forma tabellare, in WinWord (a proposito stiamo provando la versione 2.0) vanno poste in forma testuale, con i vari elementi (anche qui sette) separati da virgole. Anche in WinWord è quindi possibile realizzare Dialog Box, utilizzando oggetti Windows, e ricorrendo all'uso di vere e proprie variabili (equivalenti alla settima colonna), necessarie sia per definire i valori assunti all'inizio nella Dialog Box, sia per prelevare i valori impostati dall'utente e riversarli nel documento.

delle tante modalità permesse nei fogli elettronici (assoluto, relativo, nel foglio attivo, in un foglio specifico, ecc.).

Per quanto riguarda i Campi codificati (CheckBox, ListBox e OptionGroup) citati prima, i valori numerici potranno essere facilmente scodificati con una funzione del tipo SE, SCELTA, INDICE o SCANSVER.

Cogliamo l'occasione per precisare che il nostro obiettivo è limitato alla costruzione e all'utilizzo «interno» della Finestra. Diamo solo alcuni cenni di cosa deve succedere prima e cosa dopo il suo uso.

La Dialog Box sviluppata con WinWord 1.1

Tutte le «cose» viste in Excel esistono anche in WinWord. Esiste quindi la Finestra di Dialogo e conseguentemente esistono le istruzioni per definirne e per gestirne il contenuto.

Anche in WinWord la Finestra di Dialogo deve essere lanciata dall'interno di una Macro. Le Macro di WinWord vanno scritte in WordBasic, un linguaggio misto tra Windows e Basic, molto sofisticato e potente, ma poco documentato.

Infatti a differenza dell'Excel 3, che nella sua manualistica cita, anche se in maniera sommaria, le istruzioni di programmazione con cui si genera una Finestra di Dialogo, WinWord non dispone di un manuale relativo alle Macro. Per cui anche la semplice spiegazione dei comandi WordBasic viene rimandata ad altri manuali non forniti con il prodotto, ma disponibile solo a richiesta, come il Technical Reference, interessante, ma anche lui troppo sintetico. Alle finestre di dialogo sono dedicate solo poche pagine.

Peccato! Perché il WordBasic, che ovviamente permette di gestire variabili e vettori, permette di generare cicli e dialogare con file di dati, ben si presta ad un lavoro, anche pesante, di programmazione, e quindi una documentazione più esauriente avrebbe comunque reso meno difficile il compito a chi volesse affrontare questa attività.

Nella figura 8 vediamo la Dialog Box sovrastante il documento in cui i dati digitati nella Dialog Box stessa vanno a finire. Nella successiva figura 9, che mostra il listato della Macro AutoNew, si può notare come sia la ListBox che l'OptionGroup possano essere facilmente ed utilmente scodificati con un Vettore.

L'uscita viene realizzata con i due Bottoni OK e Annulla, mentre non è possibile eseguire il calcolo internamente alla Finestra. Il calcolo tra Importo e Percentuale viene quindi eseguito solo in uscita dalla Box, contestualmente all'Inserimento dei dati nel documento.

L'Inserimento dei dati nel documento può avvenire in svariati modi. Nel nostro esercizio abbiamo riportato i due più semplici.

Il primo è quello che fa uso dell'istruzione Inserisci, comoda, ma che non permette il posizionamento del dato in un punto fisso del documento.

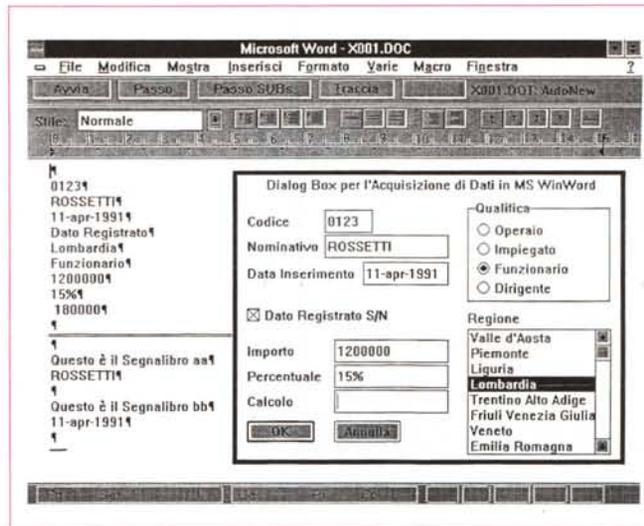
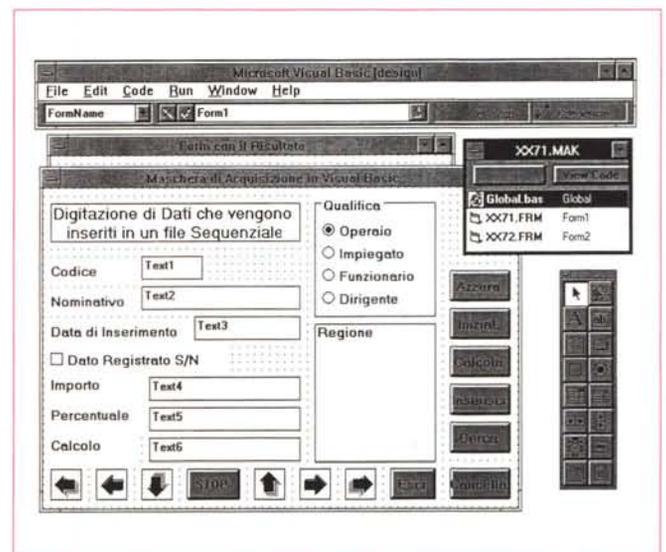


Figura 9 - Dialog Box in Windows 3 - Utilizzo in un documento WinWord della Macro che attiva la Dialog Box. Nel nostro esempio abbiamo inserito la Dialog Box nella Macro AutoNew di un Modello WinWord. Quando si inizia un nuovo documento che segue tale Modello la Macro viene eseguita automaticamente e quindi viene attivata anche la Dialog Box in essa contenuta. I dati immessi nei vari elementi della Box vanno a finire in variabili già inserite nel documento che così viene alimentato attraverso la Dialog Box stessa.

Figura 10 - Dialog Box in Windows 3 - La Dialog Box in Visual Basic. La costruzione della nostra Dialog Box in Visual Basic diventa un «affare» di tipo prevalentemente grafico. Si utilizza il Toolbox, una barra con le varie icone che rappresentano ciascuna uno degli oggetti da inserire, e si opera su un Form inizialmente vuoto, corrispondente ad un foglio da disegno quadrato. L'insieme si «anima» inserendo routine con i programmi al di sotto degli elementi attivi della composizione, ad esempio al di sotto dei Bottoni e delle icone operative.



Il secondo invece comporta il posizionamento del cursore all'interno di un testo già scritto, e nel successivo inserimento. Per posizionarsi con l'istruzione VAL.A occorre aver definito nel testo un Segnalibro. Nella nostra figura il Segnalibro si vede, in un caso reale andrà definito invece un Segnalibro invisibile, costituito ad esempio da un solo carattere spazio.

Una terza e una quarta possibilità, che però non presentiamo, consistono nell'utilizzare nel documento un testo già scritto all'interno dei quali sono stati inseriti dei campi il cui contenuto sarà costituito dai valori digitati nella Dialog Box. Ad esempio un campo di tipo {RIF SS} o uno di tipo {GLOSSARIO GG} fanno apparire come risultato o il contenuto del segnalibro SS o il contenuto della voce di glossario GG.

Questa strada è praticabile se si riesce a trasferire il valore delle variabili usate nella finestra di dialogo, rispettivamente a Segnalibri o a Voci di Glossario.

È inutile precisare che il prerequisito necessario per poter programmare efficacemente con le Macro dei vari prodotti è la

conoscenza approfondita di tutte le loro funzioni normali.

Note a margine delle Dialog Box di Excel e WinWord

A parte le logiche e prevedibili differenze tra Excel e WinWord, va notato un notevole allineamento tra i due per quanto riguarda la programmabilità e lo sfruttamento delle Finestre di Dialogo, nel caso di Excel utilizzabili per gestire un vero e proprio Data Entry, strutturato e controllato, nel caso di WinWord utilizzabile per personalizzare un documento standard.

Tornando poi alle differenze tra i due a ben pensarci non sono così evidenti se si considera che un Vettore del WordBasic coincide con una serie di celle di un foglio elettronico, e che una variabile, definibile come Segnalibro, del primo diventa nel secondo una singola cella.

Considerando infine che Excel e WinWord sono in fondo dei prodotti destinati principalmente ad utenti non specialisti i risultati che si ottengono con lo sfruttamenti

Figura 11 - Dialog Box in Windows 3 - La programmazione degli Eventi in Visual Basic. Non essendo la nostra finalità quella di predisporre un programma completo, e comunque non potendo per mancanza di spazio pubblicare tutti i listati relativi ai singoli bottoni, ne presentiamo solo un campionario. Vediamo ad esempio le istruzioni globali, che nel nostro caso servono solo per dichiarare alcune variabili globali, e le istruzioni generali del Form che servono per dichiarare i Vettori che, dovendo essere utilizzati in più sottoprogrammi interni al Form stesso, vanno dichiarati a livello generale.

```

Rem Stralci dei Sottoprogrammi richiamati da alcuni Bottoni
Rem Inseriti nella FORM Visual Basic

Rem Dichiarazioni Globali valide per tutta la Procedura
Global t4, t5, t6

Rem Dichiarazioni Generali valide nel Singolo FORM
Dim a$(19), b$(3)

Rem Bottone Inizializza
Sub Inizializza_Click ()
a$(0)="Valle d'Aosta":a$(1)="Piemonte":a$(2)="Liguria":a$(3)="Lombardia"
a$(4)="Trentino Alto Adige":a$(5)="Friuli Venezia Giulia":a$(6)="Veneto"
a$(7)="Emilia Romagna":a$(8)="Toscana":a$(9)="Marche":a$(10)="Umbria"
a$(11)="Lazio":a$(12)="Abruzzo":a$(13)="Molise":a$(14)="Campania"
a$(15)="Basilicata":a$(16)="Puglia":a$(17)="Calabria"
a$(18)="Sicilia":a$(19)="Sardegna"
b$(0)="Operaio":b$(1)="Impiegato":b$(2)="Funzionario":b$(3)="Dirigente"
text1.text="" : text2.text="" : text3.text=""
text4.text="" : text5.text="" : text6.text=""
For i = 1 To 19: regione.AddItem a$(i) : Next i
End Sub

Rem Bottone Calcola
Sub Calcola_Click ()
ft$="###,###,###,###"
t4=Val(text4.text):t5=Val(text5.text):t6=t4 * (100-t5)/100
text4.text=Format$(t4,ft$):text5.text=Format$(t5,ft$):text6.text=Format$(t6,ft$)
End Sub

Rem Bottone Inserisci
Sub Inserisci_Click ()
Load form2:form2.Show:Cls
t1$=text1.text:t2$=text2.text:t3$=text3.text
form2.Print "Questa è il sottoprogramma collegato al Bottone Inserisci"
form2.Print
form2.Print "Al nominativo "+t2$+" e' stato attribuito il codice "+t1$
form2.Print "Appartiene alla Regione "+regione.text;
form2.Print " e la sua qualifica è ";
If oper.value Then
form2.Print "Operaio"
Elseif impi.value Then
form2.Print "Impiegato"
Elseif funz.value Then
form2.Print "Funzionario"
Elseif dir1.value Then
form2.Print "Dirigente"
End If
form2.Print "Il dato ";
If check1.value = True Then
form2.Print "non è ";
Else
form2.Print "è ";
End If
form2.Print "stato registrato in data "+t3$
form2.Print "L'importo digitato è Lit. "+Str$(t4)
form2.Print "che con lo sconto applicato del "+Str$(t5)+" %";
form2.Print "diventa "+Str$(t6)
form2.Print:MsgBox "Inserimento Competato":form2.Hide:form1.Show
End Sub

```

to delle possibilità offerte dalle Dialog Box personalizzate sono senza dubbio interessanti.

Nei prodotti più evoluti, che sono ovviamente anche prodotti destinati ad utenti ben più specializzati, la Dialog Box diventa uno strumento ancora più potente e sofisticato, in grado cioè di assumere su di sé tutte le incombenze relative alla immissione, alla validazione e al calcolo dei dati necessari al buon funzionamento dell'applicazione.

Anzi, i due prodotti dei quali stiamo per parlare, che sono Microsoft Visual Basic e Borland Object Vision, ruotano completamente attorno alle Dialog Box, che nel primo

assumono il nome di Form e nel secondo quello di Scheda.

Una delle caratteristiche più importanti che differenziano le Dialog Box realizzate con uno di questi due strumenti rispetto a quelle viste prima, realizzate con Excel o WinWord, è che in esse è possibile programmare azioni che vengono eseguite anche all'interno della Dialog Box stessa.

In altre parole in Excel o in WinWord non è possibile che un calcolo venga eseguito durante la digitazione dei dati all'interno della Dialog Box, ma lo si può eseguire solo in uscita dalla Box. Con Visual Basic e Object Vision invece è possibile definire eventi al-

l'interno della stessa Dialog Box, ad esempio per eseguire un calcolo interno alla Box, oppure per attivare «al volo» una sottostante Dialog Box per l'immissione di dati più specifici, oppure per lanciare un sottoprogramma, e così via.

Anticipata questa differenza «filosofica», passiamo alla realizzazione della nostra Dialog Box con il Visual Basic.

La Form in Visual Basic

Nel trattare il Visual Basic diamo per note le sue principali caratteristiche o perlomeno per «letti» i due articoli apparsi su MC numero 111 e dedicati a tale prodotto.

In Visual Basic la Form, alias Dialog Box, si disegna direttamente con gli strumenti grafici a disposizione scegliendo e posizionando oggetti presenti nella ToolBox (fig. 10).

Di ogni oggetto vanno stabilite le «Property» e per ciascuna di queste i relativi parametri. Property e parametri possono anche essere modificati da programma in quanto ad ogni operazione eseguita via menu, o con gli strumenti di disegno, corrisponde anche e sempre un comando di programmazione.

Per ogni oggetto inserito in Form possono essere individuati degli eventi (nessuno, uno, più di uno), al verificarsi dei quali viene eseguito un sottoprogramma che l'utente lega a quell'evento (programmazione Event Driven).

La conseguenza, oltre al fatto che è possibile eseguire sottoprogrammi senza uscire dalla Form, è che i sottoprogrammi stessi sono legati indissolubilmente all'oggetto che li manda in esecuzione (Programmazione ad Oggetti).

In questo modo l'attività di programmazione viene semplificata in quanto invece di realizzare un unico lungo programma con tante subroutine e quindi con tanti richiami e ritorni, basta programmare il singolo evento previsto senza preoccuparsi troppo delle altre subroutine legate agli altri eventi.

Per dare il massimo risalto a questo concetto fondamentale abbiamo introdotto nella nostra Dialog Box tantissimi Bottoni, anche di tipo grafico, a ciascuno dei quali abbiamo legato una singola azione elementare sui dati della nostra Dialog Box.

Supponendo quindi che la Dialog Box serva per visualizzare e gestire i record di un archivio, abbiamo «bottonizzato» sia le operazioni di scorrimento necessarie per mostrare via via i record, sia le operazioni di aggiornamento dei dati di un singolo record.

Le prime le abbiamo legate ad una pulsantiera grafica tipo videoregistratore, facilmente realizzabile utilizzando alcune delle Icone con le Frecce, scelte tra le centinaia a disposizione nel materiale fornito con il Visual Basic (Avanti, Indietro, Avanti Veloce, Indietro Veloce, Stop, Vai al Primo, Vai all'Ultimo), le altre che, come detto, servono per compiere operazioni di gestione del record, sono costituite da Bottoni più semplici.

Non essendo la nostra finalità quella di

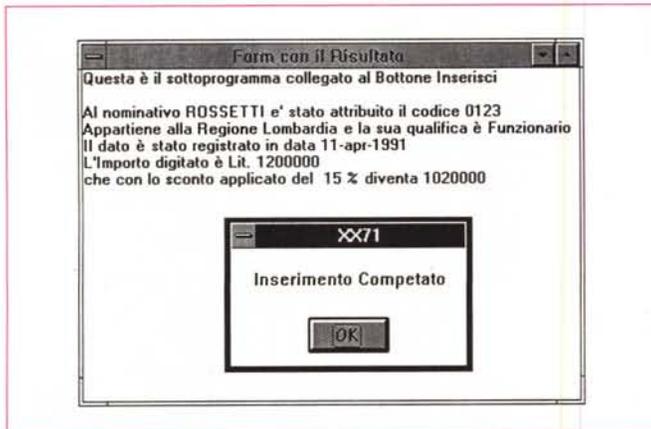


Figura 12 - Dialog Box in Windows 3 - L'output su un nuovo Form in Visual Basic. Questa figura mostra l'Output legato all'evento Inserisci, che prevede l'attivazione di un secondo Form vuoto nel quale, con una serie di istruzioni di tipo Print, viene riversato il contenuto delle variabili gestite nel primo Form. In una reale procedura tale contenuto andrà invece a finire in archivi su disco.

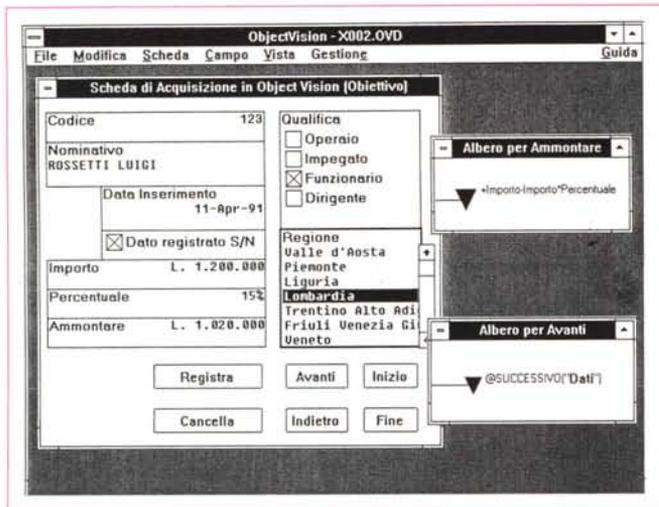


Figura 14 - Dialog Box in Windows 3 - La Dialog Box in ToolBook 1.x.

Nello «STUDIO» dell'ambiente Windows assume una notevole importanza l'Asymetrix ToolBook, cui si potrebbe assegnare l'appellativo di «Prodotto di Testo». ToolBook infatti contiene, e soprattutto documenta efficacemente nella voluminosa manualistica, tutte le funzionalità tipiche di Windows. Permette di realizzare Dialog Box, menu personalizzati, di attivare Bottoni di vario tipo, di lavorare con le DLL, ecc. ecc.



predisporre un programma completo, e comunque non potendo per mancanza di spazio pubblicare tutti i listati relativi ai singoli bottoni, ne presentiamo solo una cospicua rappresentanza in figura 11. In particolare vediamo le istruzioni globali, che nel nostro caso servono solo per dichiarare alcune Variabili Globali. Vediamo poi le istruzioni generali della Form che servono per dichiarare i Vettori che, dovendo essere utilizzati in più sottoprogrammi interni alla Form, vanno dichiarate a livello generale.

Il bottone Inizializza. Serve per inizializzare le variabili (alcune vanno svuotate, altre settate al loro valore di Default) e per riempire la ListBox con il suo contenuto e va eseguito prima della digitazione dei Dati.

Il bottone Calcola. Serve per eseguire i calcoli interni alla Dialog Box, che nel nostro caso si limitano alla moltiplicazione tra i due valori Importo e Percentuale con la quale si ottiene il valore calcolato dell' Ammontare.

Il bottone Inserisci. Serve per illustrare un semplice utilizzo dei dati digitati nella Dialog

Box. In pratica viene attivata una Form vuota nella quale con l'istruzione Print, in collaborazione con alcune Funzioni, viene riversato il contenuto della Dialog Box. Abbiamo preferito questa soluzione più semplice rispetto ad un più logico uso delle funzioni di gestione dei File, per non appesantire troppo la trattazione.

Ricapitolando. Delle tre figure dedicate al Visual Basic la prima mostra la Form in lavorazione. La seconda è costituita da alcuni listati relativi agli eventi previsti quando si clicca su alcuni dei Bottoni. La terza, appena citata, mostra l'output legato all'evento Inserisci, che prevede l'attivazione di una Form vuota nella quale con una serie di istruzioni di tipo Print viene riversato il contenuto delle variabili gestite nella Form.

La scheda con Borland Object Vision

L'Object Vision serve proprio per realizzare Dialog Box, intese come «visione» degli

Figura 13 - Dialog Box in Windows 3 - La Dialog Box in Object Vision.

In un certo senso l'Object Vision della Borland costituisce un generatore «intelligente» di Dialog Box «intelligenti». Nella scheda generata con OV possono non solo essere inseriti gli oggetti standard Windows, ma anche routine di calcolo e di controllo, che servono per filtrare ed elaborare i dati immessi, e regole di collegamento in modo tale che i dati immessi siano riversati in un file esterno preesistente disponibile in un formato standard.

«oggetti». Gli oggetti anche in questo caso appartengono alla famiglia Windows, anche in questo caso rappresentano dei dati disponibili su archivi e anche in questo caso non vanno programmati in quanto le loro caratteristiche vanno semplicemente impostate attraverso una serie di Dialog Box operative da riempire.

L'oggetto può anche comprendere una parte testuale fissa e questo comporta una ulteriore semplificazione delle operazioni di disegno della Dialog Box. Non esiste un linguaggio di programmazione vero e proprio, esiste una sorta di linguaggio grafico, che utilizza elementi grafici abbastanza simili alle simbologie della diagrammazione a blocchi, e con il quale si impostano sia le «formule» dei campi calcolati, anche nel caso di calcoli complessi che comprendono dei salti, sia le «formule» che legano il contenuto della Dialog Box ad un file esterno.

Finito il primo tempo, costituito dal disegno della Dialog Box e dalla definizione del suo contenuto, si passa alla creazione del collegamento tra i campi inseriti nella Dialog Box stessa e l'archivio sottostante. Anche il formato di tale archivio è «a scelta» nel senso che durante l'operazione di Link appaiono una serie di Dialog Box in cui si decide formato, nome e campi degli archivi da collegare.

Nella figura 13 vediamo la nostra Dialog Box e due piccole finestre che mostrano la nostra moltiplicazione e una delle funzioni «chiocciolina» che eseguono operazioni sull'archivio esterno (può essere dBase III, Paradox, DDE, ASCII) «legato» alla Dialog Box. Ad esempio la funzione @successivo("dati") provoca uno «skip» al record successivo dell'archivio con il quale ci si è collegati.

Non è necessario utilizzare dei Bottoni per lanciare i calcoli in quanto questi sono comunque rieseguiti quando si varia uno dei parametri implicati nel calcolo stesso.

Nell'esempio Visual Basic visto prima abbiamo invece inserito un Bottone Calcola, per produrre un calcolo. Potevamo evitarlo, gestendo non l'evento Click, ma gli eventi Change di uno degli oggetti coinvolti nel calcolo.

Asymetrix ToolBook 1.5 e 1.0

Nello «STUDIO» dell'ambiente Windows diamo notevole importanza all'Asymetrix ToolBook, cui si potrebbe assegnare l'appellativo di «Prodotto di Testo», mutuando tale termini da quello di «Libro di Testo» vigente in tutte le altre attività di Studio. Di ToolBook abbiamo parlato più volte e quindi diamo ora per scontata un minimo di conoscenza acquisita leggendo tali articoli.

ToolBook contiene, e soprattutto documenta efficacemente nella voluminosa manualistica, tutte le funzionalità tipiche di Windows. Tali funzionalità sono attivabili sia dall'ambiente operativo interattivo cui accede l'«Author» del Book, sia dall'ambiente programmazione, che anche in que-

Figura 15 - Dialog Box in Windows 3 - Utilizzo delle DLL del ToolBook.

I dati acquisiti con la Dialog Box sono destinati a finire da qualche parte, ad esempio nel foglio di calcolo, se parliamo di Excel, nel documento se lavoriamo con WinWord, in un file Sequenziale o Random, se lavoriamo con Visual Basic, in un file DBF, Paradox o in un canale DDE, se lavoriamo con Object Vision. L'ultimo prodotto in esame, il ToolBook, può sfruttare la sua dotazione di librerie DLL, ad esempio questa, che dispone di una trentina di funzioni corrispondenti ad altrettanti comandi dBase III.

```

to handle linkdBase
-- Collegamento alla libreria dBase dll e dichiarazioni delle Funzioni utilizzabili
clear sysError
set sysSuspend to false
linkdll "tbkdb3.dll"
-- Funzioni operanti sui files
INT packDBFile()
INT opendirFile(STRING)
INT closeDirFile(STRING)
INT closeAllDirFiles()
INT createDBIndexFile(STRING,STRING,WORD,WORD)
INT openDBIndexFile(STRING)
INT deSelectDBIndexFile()
STRING getDBIndexExpression()
STRING getDBIndexFileName()
INT closeDBIndexFile(STRING)
INT selectDBIndexFile(STRING)
INT checkDBIndex(STRING)
INT firstDBKey()
INT lastDBKey()
INT nextDBKey()
INT previousDBKey()
INT findDBKey(STRING)
STRING getDBKeyValue()
-- Funzioni operanti sui record
LONG getDBRecordNumber()
LONG getDBRecordCount()
INT getDBRecordDeleted()
INT writeDBRecord(DWORD)
INT firstDBRecord()
INT lastDBRecord()
INT nextDBRecord()
INT previousDBRecord()
INT gotoDBRecord(DWORD)
INT setDBRecordDeleted(WORD)
INT setDBNavigateToDeleted(INT)
INT getDBNavigateToDeleted()
-- Funzioni operanti sui campi
STRING getDBFieldValue(STRING)
INT setDBFieldValue(STRING,STRING)
INT getDBFieldType(STRING)
-- Interpretazione degli Errori
STRING getDBErrorString(INT)
end linkdll

-----

to handle writeRecord fType
system svCurrentRecord, svTotalRecords, svFieldNames, svTotalRecords
local vThisField
send updateWarning
-- Controlla se il record corrente e' marcato per la cancellazione
get getDBRecordDeleted()
if it = 1 and fType is not "New"
-- Visualizzazione di un messaggio di errore
request strInvalidRecord()
-- Comando di Uscita
break to system
end if
if fType is "New"
-- Azzeri i campi
send clearFields
end
if fType is "New"
-- Inserisce nella variabile svTotalRecords il numero dei record
-- presenti nell'archivio
set svTotalRecords to getDBRecordCount()
-- Registra il record posizionandosi sull'ultimo (Append Blank)
get writeDBRecord(svTotalRecords+1)
-- Incrementa la variabile
increment svTotalRecords
-- Riscrive sopra il record corrente
get writeDBRecord(svCurrentRecord)
end
-- Visualizza il record corrente
send updateDisplay
end writeRecord

```

sto caso è Event Driven. La costruzione della Dialog Box si può eseguire in maniera grafica, in modo abbastanza analogo a quelli descritti fino ad ora, sia nell'ambiente normale di programmazione sia ricorrendo ad una specifica applicazione Dialog Box Editor (fig. 14).

Quello che ci stimola particolarmente è la possibilità di gestire direttamente il rapporto tra Dialog Box ed archivi sottostanti... lavorando direttamente in dBase III, o meglio sfruttando la dotazione di ToolBook di librerie DLL, ed in particolare utilizzando quella che dispone di una trentina di funzioni corrispondenti ad altrettanti comandi dBase III. Delle DLL parleremo in uno dei prossimi numeri di MC.

In figura 15 vediamo uno stralcio dei programmi sottostanti la finestra di dialogo, ed in particolare la sezione in cui vengono dichiarate le varie funzioni con le relative specifiche per il dialogo con file DBF, presenti nella libreria TBKDB3.DLL, e un assaggio del programma legato all'evento che genera la scrittura del Record.

ToolBook fortunatamente documenta molto bene la sintassi delle funzioni di tale libreria (fig. 16) per cui si pone come interfaccia ideale per chi debba avere una estrema libertà non tanto nel disegnare la Dialog Box quanto per gestire i rapporti tra gli oggetti in essa contenuti e... i dati, dovunque risiedono e in qualsiasi formato siano disponibili.

Conclusioni

Notiamo subito la differenza tra la soluzione ToolBook, più tecnica, destinata al programmatore, che se vuole parlare con file DBF, deve necessariamente affrontare le DLL, e la soluzione Object Vision, in cui il rapporto tra Front End (quello che vede l'utente) e i dati è mediato da semplici funzioni chioccioline.

Ma al di là della ricchezza delle «varianti» al problema della costruzione e della programmazione delle Dialog Box, varianti che servono a definire principalmente i confini dell'area di applicabilità di ciascun prodotto, quello che ci interessa notare è come oggi la Dialog Box sia diventata l'interfaccia standard tra utente e applicazione e come conseguentemente il suo uso sia diventato un «patrimonio culturale» comune, nel senso che chiunque abbia usato Windows e un prodotto Windows sa intrinsecamente usare una Dialog Box. E anche chi usa un prodotto Windows per la prima volta riesce facilmente a capirla.

Nei prossimi anni la Dialog Box non varierà, varierà quello che le sta dietro, in quanto usciranno tool sempre più sofisticati che permetteranno in maniera sempre più semplice di costruire il suo Back End anche di tipo evoluto. Ed è probabile che tale semplicità sarà consentita soprattutto dall'uso massiccio di Dialog Box e di DLL che evitino all'utente il contatto sempre rude ed impegnativo con il linguaggio, quale esso sia.

MS

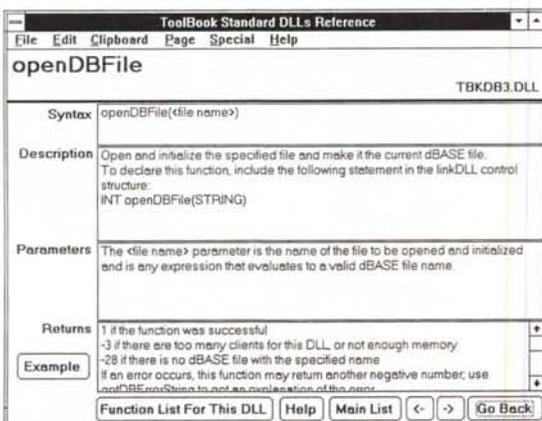


Figura 16 - Dialog Box in Windows 3 - La sintassi di una Funzione della Libreria TBKDB3.DLL.

Come noto Windows ha proprie librerie dinamiche richiamate al volo sia dagli applicativi normali che dalle applicazioni scritte dagli utenti. Molti prodotti, per proprie necessità, aggiungono a tale dotazione le proprie DLL che diventano così «patrimonio» comune dell'ambiente Windows e possono essere sfruttate da qualsiasi altro prodotto che abbia proprie istruzioni di tipo CALL.

