

# Microsoft Visual Basic

## Costruzione ed utilizzo di OLE Server

La Programmazione per Componenti (Object Based oppure Component Based) è diventata, in questi anni, il modo di programmare standard per chi deve costruire applicazioni, soprattutto di tipo gestionale, in ambiente Windows.

Con l'ultima valanga di prodotti, causata dall'uscita di Windows 95 e soprattutto dalla diffusione delle tecnologie legate al fenomeno Internet/Intranet, questa modalità di programmazione si è ulteriormente e definitivamente consolidata.

È chiaro che, come in ogni tecnologia, anche nella programmazione Component Based esistono aspetti teorici (che in questo caso sono fortunatamente pochi) e aspetti pratici (che invece sono tantissimi).

Come al solito faremo una puntatina sui primi e ci butteremo "a pesce" sui secondi.

### Prima parte

#### Cosa faremo

Per Programmazione a Componenti intendiamo la costruzione di oggetti che svolgono una serie di funzionalità proprie, che sono poi facilmente richiamabili da qualsiasi altra applicazione, che li utilizza come se fossero una propria parte.

Rifacendo questo stesso discorso, riferendoci direttamente all'ambiente Windows, che è quello che ci interessa, parleremo di componenti OLE Server, la definizione dei quali è la seguente: un OLE Server è un'applicazione che espone proprie funzionalità che possono essere usate da altre applicazioni, che nell'occasione si comportano come OLE Client, attraverso la tecnologia OLE Au-

tomation.

Il vantaggio di lavorare per componenti è quindi quello di mettere "a fattor comune" le parti di codice più utilizzate rendendole disponibili non solo dall'interno della stessa applicazione ma anche da parte di varie altre applicazioni. Questo vantaggio diventa ancora più sensibile quando si sfrutta la tecnica chiamata Remote Automation, nella quale l'applicazione OLE Server risiede ed opera su un server, ed è richiamata da varie Applicazioni e da vari client.

In questo caso si abbattano drasticamente i costi di manutenzione del codice in quanto una modifica, apportata al solo componente che risiede sul server, diventa immediatamente disponibile ai vari client, sui cui possono girare anche

applicazioni differenti.

Per sperimentare questi concetti lavoreremo con il Visual Basic, che useremo sia per costruire alcuni OLE Server, sia per utilizzarli. Lavoreremo anche con Excel con il quale sfrutteremo gli OLE Server realizzati con VB.

Alla fine faremo anche il contrario, partendo dal Visual Basic, che si può comportare come un qualsiasi OLE Client, useremo l'intero Excel come OLE Server.

In questo primo articolo sull'argomento "Costruzione ed Utilizzo di OLE Server" ci attestiamo su Visual Basic 4.0, in inglese, e su Office 95. In successivi articoli, quando sarà effettivamente disponibile e opportunamente documentato, vedremo Visual Basic

5.0, che in questo particolare aspetto (costruzione di componenti) è molto più evoluto del precedente, in quanto consente di costruire non solo OLE Server, ma anche oggetti OCX e ActiveX.

Passando agli aspetti pratici dell'articolo, vi proporremo tre esercizi, ognuno dei quali vale per due, nel senso che costruiremo sia la parte server che la parte client, o per tre, se, ad esempio, useremo come parte client anche l'Excel. I tre esercizi sono:

- 1) la costruzione di una semplice funzione inserita in un OLE Server,
- 2) la costruzione di un OLE Server, che usa una sua form, che riceve dati dal client e che restituisce dati al client,
- 3) l'esempio numero 3 non c'è, e spieghiamo dopo il perché,
- 4) la costruzione di un OLE Server, che usa una sua form, che riceve collezioni di dati dal client e restituisce collezioni di dati al client.

Useremo gli OLE Server, realizzati con il primo e con il quarto esercizio, anche da Excel.

Il terzo esercizio, quello che non c'è, serve come preparazione del quarto. In pratica realizzeremo un'unica applicazione che usa due form, ben funzionante soprattutto per quanto riguarda l'apertura e la chiusura delle form e il passaggio dei dati, anche collezioni di dati, da una form all'altra. Terminata l'applicazione, e verificato il suo corretto funzionamento, la spezziamo in due, la prima, che sarà l'OLE Client, con la prima form, e la seconda, con la seconda form, che diventerà il nostro OLE Server.

In pratica quindi il quarto esercizio consiste nella scomposizione dell'applicazione realizzata con il terzo.

## Due o tre cose sugli OLE Server

Abbiamo detto che un OLE Server è un'applicazione che espone propri oggetti che possono essere programmati da altre applicazioni attraverso l'interfaccia OLE Automation.

Quando si realizza un OLE Server non si realizza un oggetto OLE ma si realizza una "Classe".

Una classe è un "Modello", un prototipo di un oggetto. L'applicazione client non usa direttamente l'oggetto OLE ma ne crea ed usa una "Istanza". Si posso-

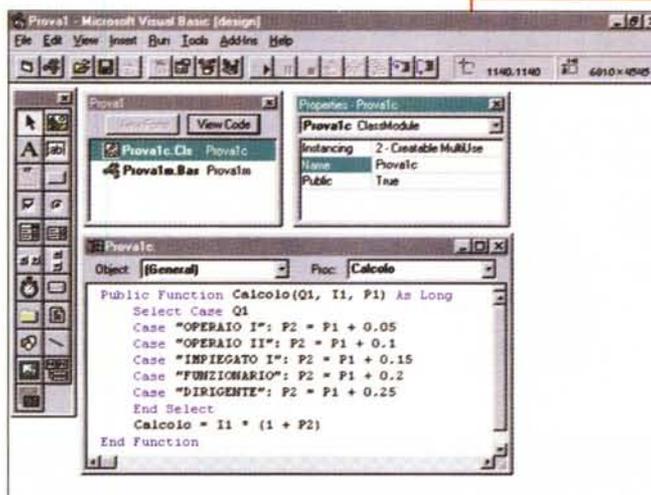


Figura 2 - Creazione di server OLE con MS VB - Utilizzo della funzione in un programma DAO.

La nostra funzione necessita di tre parametri (una qualifica, un importo ed una percentuale) e, sulla base di questi tre parametri, esegue un semplice calcolo, che è il valore restituito della funzione stessa. La utilizziamo in un programma che scorre una tabella di un database e che, con i valori presenti nei suoi vari record, esegue il calcolo. È evidente che l'unica differenza tra una funzione tradizionale, costruita in un modulo Visual Basic, e la stessa funzione, inserita in una libreria DLL, consiste nel fatto che, in questo secondo caso, la funzione diventa facilmente utilizzabile da qualsiasi applicativo client.

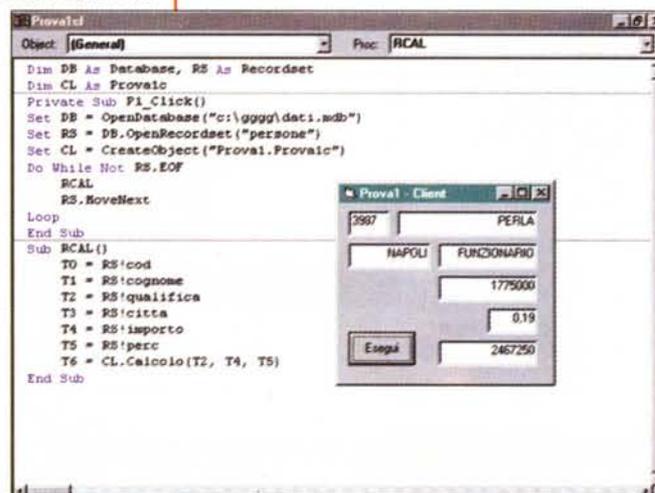


Figura 1 - Creazione di server OLE con MS VB - Costruzione di una funzione in un modulo di classe.

Come al solito, quando trattiamo un argomento tecnico, procediamo con una serie di esercizi di crescente difficoltà, che possono essere eseguiti anche da utilizzatori non particolarmente esperti. Il primo consiste nella costruzione di un OLE Server, che poi salveremo come libreria DLL, che contiene una sola e semplice funzione. Una volta compilato l'OLE Server, potremo richiamare la funzione in esso contenuta non solo da un'altra applicazione Visual Basic ma anche da qualsiasi applicazione client, ad esempio da un'applicazione Excel.

no creare ed utilizzare contemporaneamente più istanze dello stesso oggetto OLE.

Il paragone più calzante, per spiegare cosa è un OLE Server ricorrendo ad una similitudine, è quello che assimila la classe ad uno stampo, e ciascuna istanza ad uno dei numerosi elementi ottenibili dallo stampo.

Per realizzare un OLE Server occorre, al minimo, un modulo di classe, che contiene le necessarie istruzioni. Un OLE Server ovviamente può utilizzare, ma non è che sia obbligatorio, una o più

form, la cui gestione avviene attraverso il modulo di classe.

Le procedure inserite nel modulo di classe corrispondono ai "Metodi", mentre le "Variabili Pubbliche" inserite nel modulo di classe corrispondono alle proprietà dell'oggetto.

Indichiamo subito le operazioni da compiere quando si sta creando un OLE Server e che differiscono da quelle da compiere quando si crea un'applicazione normale.

Si inizia un nuovo progetto e nella finestra "Tools Options Project" si indica:

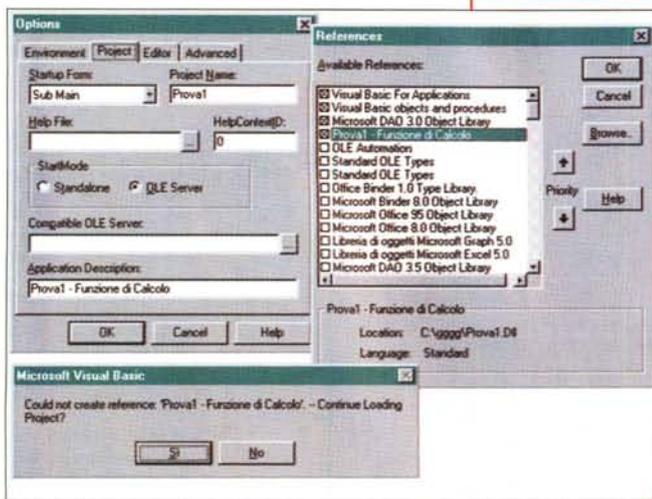


Figura 3 - Creazione di server OLE con MS VB - Passi operativi nella costruzione e nell'uso della funzione.

Questa figura è un collage costruito con tre Dialog Box, opportunamente ridotte per farle entrare nella... foto. La prima, meglio descritta nel testo, è quella che mostra le impostazioni da attivare quando si voglia realizzare un OLE Server (comando Tools Options Project). La seconda è quella che mostra l'elenco degli OLE Server disponibili nel sistema che si sta usando e che si deve attivare quando si realizza il client (comando Tools References). L'ultima, in basso, è la Message Box che indica l'errore che si verifica quando il client usa un server che non c'è (o che non è stato referenziato).

Startup form: Sub Main  
in quanto vogliamo che il nostro progetto non parta da una form ma da una procedura, che in genere è Sub Main,

ProjectName: Prova1  
nome assunto dal nostro OLE Server e con il quale viene richiamato dal client,

Start Mode: OLE Server  
per indicare che l'applicazione deve partire come OLE Server,

Application Description: Il mio primo Server OLE

commento in chiaro che appare nella box delle "Reference" dell'applicazione client.

Occorre creare, e tant'è farlo subito, un modulo BAS, che contenga la sola routine:

```
Sub Main
End Sub
```

il cui unico scopo è quello di far partire il server.

Si crea poi un modulo di classe nel quale si scrive il codice relativo a metodi e proprietà dell'oggetto che si sta creando. Il codice è relativo alle funzionalità che deve svolgere l'OLE Server e quindi lo descriveremo per ciascun singolo esercizio che faremo.

Le proprietà generali del modulo di classe rivestono una grande importanza e sono:

```
Instancing: 0 No Creatable
             1 Creatable SingleUse
             2 Creatable MultiUse
```

Determina se la classe può essere creata al di fuori del progetto oppure se ogni istanza richiesta esegua un suo differente server.

```
Name: nome della Classe ad esempio Prova1C
```

Che identifica la classe e che serve per richiamarla.

```
Public: False/True
```

Si sceglie "Public" se la classe deve essere richiamata dal client.

La classe dispone di due procedure standard, legate a due eventi standard, oltre a quelle create dal programmatore, e che sono:

```
Private Sub Class_Initialize()
End Sub
```

```
Private Sub Class_Terminate()
End Sub
```

Non è obbligatorio utilizzarle ma può essere comodo farlo. La prima corrisponde al momento in cui viene creata una nuova istanza della classe e può essere

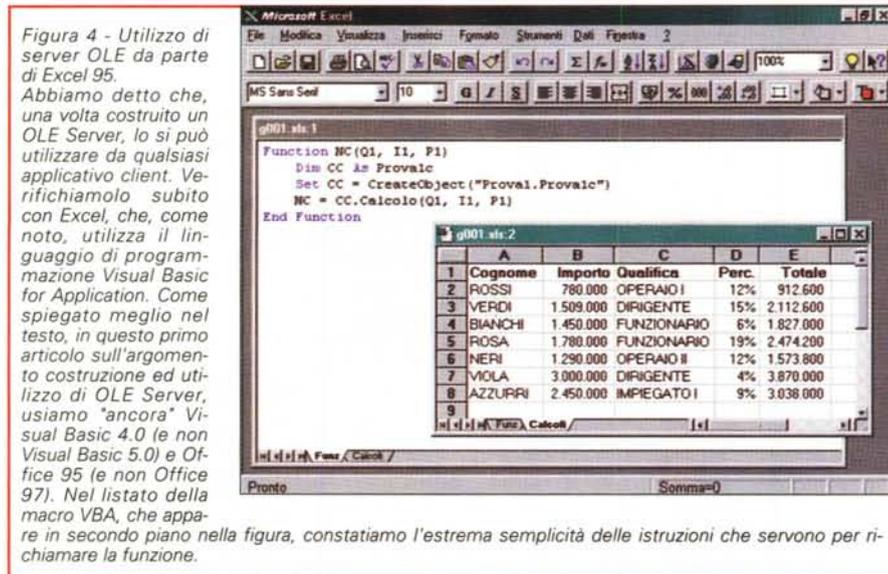
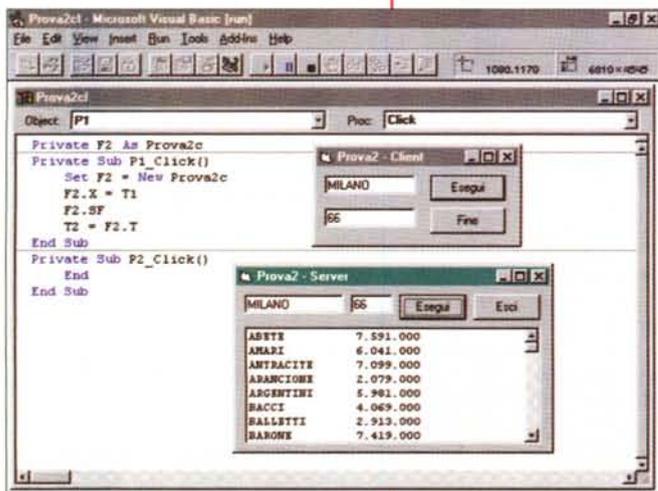


Figura 4 - Utilizzo di server OLE da parte di Excel 95.

Abbiamo detto che, una volta costruito un OLE Server, lo si può utilizzare da qualsiasi applicativo client. Verifichiamolo subito con Excel, che, come noto, utilizza il linguaggio di programmazione Visual Basic for Application. Come spiegato meglio nel testo, in questo primo articolo sull'argomento costruzione ed utilizzo di OLE Server, usiamo "ancora" Visual Basic 4.0 (e non Visual Basic 5.0) e Office 95 (e non Office 97). Nel listato della macro VBA, che appare in secondo piano nella figura, constatiamo l'estrema semplicità delle istruzioni che servono per richiamare la funzione.

Figura 5 - Creazione di server OLE che utilizza una form - Vista lato client.

In questo secondo esercizio procederemo al contrario. Vediamo prima la parte client, per capire la finalità dell'esercizio, e poi la parte server. Il client consiste in una piccola form con due TextBox e due pulsanti. Si inserisce il nome di una città nella prima TextBox e si preme "Esegui". La routine sottostante crea un oggetto OLE che si materializza in una form, nella quale si esegue una ricerca in un database, sulla base della città indicata nella form client (che fa da criterio di ricerca), la quale poi mostra il risultato della ricerca in una ListBox ed esegue un semplice calcolo, e, alla chiusura della form server, restituisce tale valore alla form client, che lo inserisce nella seconda TextBox. Il server OLE, che si chiama Prova2C, dispone di due proprietà (X, la città, T, il risultato del calcolo) e di un metodo, SF, che mostra la form.



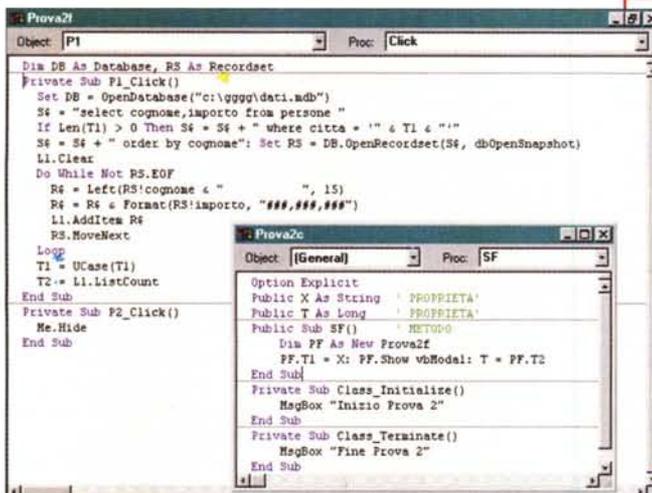


Figura 6 - Creazione di server OLE che utilizza una form - Vista lato server.

Abbiamo una certa difficoltà nel documentare i nostri esercizi per il fatto che il codice è molto spezzettato. Un po' risiede nella parte client, un po' nella parte server. Nel caso, è quello in cui ci troviamo, in cui il server utilizza una form, parte del codice risiede nel modulo della form, parte nel modulo di classe. Le figure sono quindi dei collage, e ci serve un po' di collaborazione da parte vostra per posizionare, al giusto posto, i vari "pezzetti". In questa figura vediamo sia il codice della form, sullo sfondo, sia quello del modulo di classe, in primo piano.

istanza dalla quale richiamare e quindi testare il server. Il comando è "Run Start with Full Compile".

Quando l'OLE Server è a posto lo si può compilare in due modi. Si può creare una Libreria DLL (comando File Make OLE DLL File) oppure un eseguibile (comando File Make EXE File). Ci sono restrizioni nell'uso di uno o dell'altro modo, ci sono vantaggi e svantaggi nell'uso di uno o dell'altro modo. Ne parleremo nel prossimo articolo, per ora anticipiamo il fatto che se si crea un EXE, il server "gira autonomamente" rispetto al client, se si crea una DLL, il suo uso da parte del client

è interno, quindi più diretto e più veloce. Nel primo caso si parla di Out-of-Process Server, nel secondo si parla di In-Process Server.

Passando dalla parte del client, per poter utilizzare l'OLE Server occorre selezionarlo nel pannello delle referenze con il comando "Tools References", nel quale appaiono le "Application Description" degli OLE Server. Poi le singole classi e i rispettivi metodi e proprietà possono essere raggiunti con il pannello "View Object Browser".

Ad esempio nel caso in cui il nome dell'OLE Server sia Prova1 e il nome della classe sia Prova1c, per richiamarne, dal client, una istanza, il comando sarà:

```
Dim Y As New Prova1.Prova1c
per settarne una proprietà, supposto che questa si chiami PP:
```

Figura 7 - Creazione di server OLE che riceve una Collezione e restituisce una Collezione - Output.

Per realizzare il terzo esercizio, molto più complesso dei due precedenti, procediamo per gradi. Prima realizziamo un'unica procedura in Visual Basic 4.0 (questa nella figura). Poi, verificatane la perfetta funzionalità, la spezziamo in due, la parte OLE Server e la parte OLE Client. È chiaro che tale operazione non è del tutto banale, ma la eseguiremo quando il resto dell'applicazione sarà sicuramente funzionante. Si tratta, grosso modo, di convertire alcune variabili in proprietà ed alcune procedure in metodi. La figura fa capire lo scopo dell'esercizio: abbiamo una lista, proposta in qualsiasi forma (un elenco di città in una TextBox). Di questa lista vogliamo scegliere alcuni elementi, che ci vengono restituiti in una seconda lista. Questa nella figura è la versione "unificata" dell'esercizio, che chiameremo DueListe.

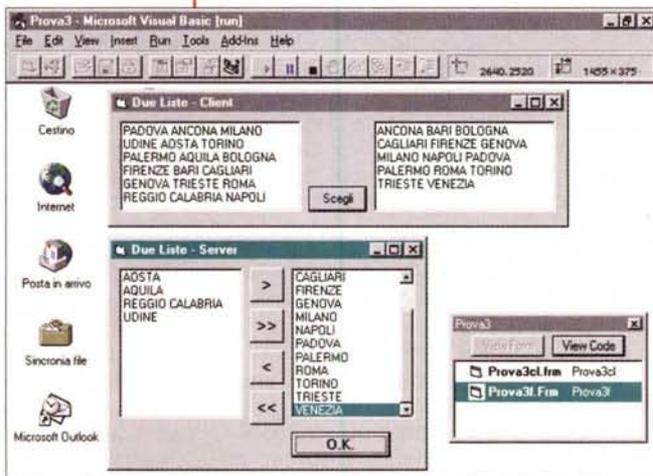
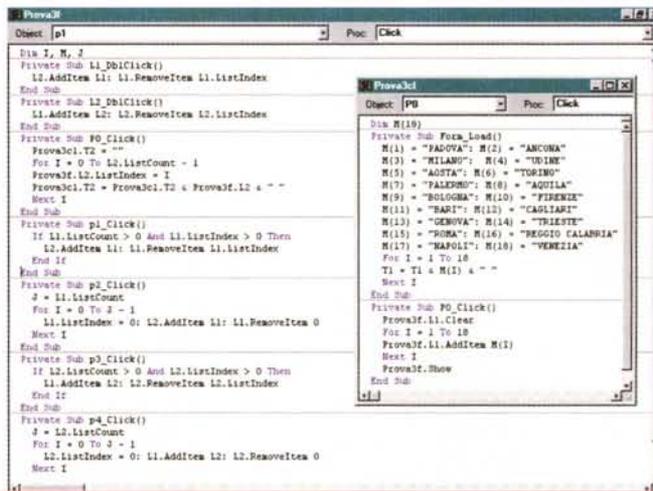


Figura 8 - Creazione di server OLE che riceve una Collezione e restituisce una Collezione - Listati.

Questo è il listato integrale dell'esercizio prima della separazione. Per le form abbiamo già adottato dei

nomi che ne indicano l'uso futuro. La form Prova3f diventerà la form Prova4f mentre la form Prova3c diventerà la form Prova4c del client. Il grosso del programma riguarda le funzionalità della seconda form (quella del server), che mostra due liste, la prima piena e la seconda da riempire con gli item scelti dalla prima. Per scegliere sfruttiamo sia i quattro pulsanti sia gli eventi "doppio click" sugli elementi della lista. L'elenco delle città da scegliere lo forniamo, banalmente, tramite un vettore di stringhe.



utilizzata per tutti i settaggi iniziali. La seconda si verifica quando l'oggetto è posto a "Nothing" (istruzione che serve per dissociare variabile ed oggetto).

Se l'OLE Server prevede l'utilizzo di una form è necessario creare un'istanza della form dall'interno del modulo di classe, in quanto il client non può referenziare direttamente la form. Un esempio di codice è:

```
Dim X As New Form1
```

Per testare l'OLE Server si può eseguire la compilazione completa, lasciare attivo il Visual Basic, e lanciaarne un'altra

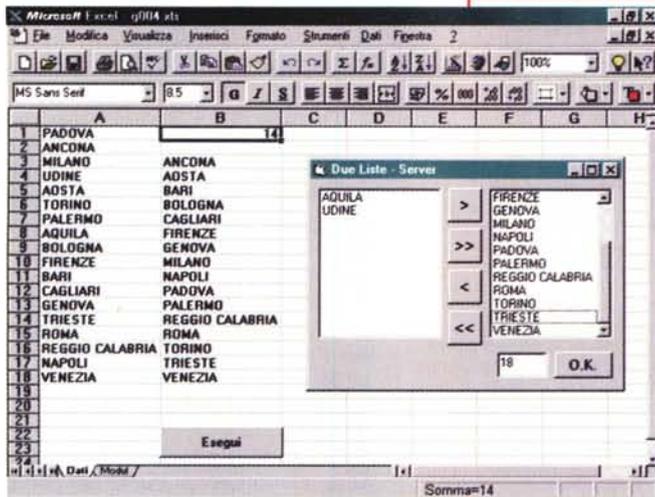


Figura 9 - Utilizzo, da Excel, dell'OLE Server DueListe - Vista dal foglio.

Il nostro esercizio Prova3 si spezzerà in Prova4 (server) e Prova4C (client). Una volta messo a punto il server lo potremo facilmente usare da Excel. Facilmente perché il grosso del lavoro lo svolge proprio il server. Qui vediamo un foglio Excel con i due elenchi, quello con le città da scegliere e quello con le città scelte. Nella cella B1 abbiamo il numero degli elementi scelti. In basso un pulsante che esegue la procedura che si chiama DueListe e che stabilisce il collegamento con il server.

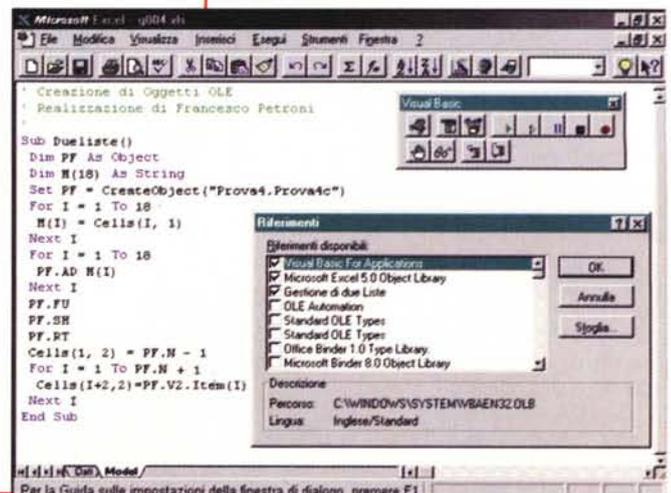
Un problema, più nostro che stiamo scrivendo l'articolo che vostro quando lavorerete, è quello di come descrivere ed illustrare i vari esercizi. Nel caso generico ci sono almeno cinque cose da vedere (che richiederebbero cinque figure):

- 1) la form del server
- 2) il codice della form del server
- 3) il codice della classe del server
- 4) la form del client
- 5) il codice della form del client.

Non potendo, per mancanza di spazio, inserire tante figure, realizzeremo dei collage, e vi indicheremo cosa contengono, fidando anche su di voi per la comprensione dei vari "pezzi".

Figura 10 - Utilizzo, da Excel, dell'OLE Server DueListe - Vista della macro DueListe.

Il listato della macro DueListe è, come detto, relativamente banale in quanto il grosso del lavoro è svolto dal server. Buona parte del listato riguarda le istruzioni che servono per trasferire i dati dall'elenco di partenza alla prima lista dell'OLE Server, per mezzo del metodo AD applicato all'oggetto PF. Il ritorno invece è ottenuto sfruttando la collezione V3 presente nel server. Una collezione, in Visual Basic, è un insieme di elementi organizzati e gestibili con una serie di metodi, ed è assimilabile.



## Un OLE Server come Libreria di Funzioni

Un OLE Server può svolgere una serie di servizi, il più semplice dei quali è quello di fungere da Libreria di Funzioni. Vediamo come si fa, riferendoci a quanto mostrato in figura 1.

La nostra applicazione necessita di un modulo, in cui scrivere la routine Sub Main, il cui solo scopo è quello di far partire il tutto, e di un modulo di classe, in cui scrivere le istruzioni per la nostra funzione. La funzione esegue un certo calcolo, ad esempio il calcolo di uno stipendio, sulla base di una serie di parametri che le vengono passati. Nel caso specifico le passiamo tre parametri e la funzione poi restituisce un valore numerico.

Le istruzioni necessarie per il calcolo del risultato le vediamo in figura 1.

Poi vanno definite le proprietà del modulo di classe (anche questo si vede in figura 1), secondo quanto specificato nella parte iniziale.

In questo caso la cosa più indicata è la creazione di una libreria DLL, magari contenente una serie di funzioni di calcolo riguardanti il calcolo degli stipendi.

Nella seconda figura vediamo come sia possibile usare la funzione da un'applicazione client.

Si tratta di un'applicazione che apre un database ed una tabella, ne scorre i dati e, sulla base dei valori di tre dei campi, esegue il calcolo.

Le istruzioni necessarie per referenziare e creare l'oggetto OLE sono:

```
Dim CL As Prova1.Prova1c
Set CL = CreateObject("Prova1.Prova1c")
per attivare la funzione:
```

Y.PP = valore  
per attivarne un metodo, supposto che questo si chiami MM:

Y.MM

Se il componente OLE Server è "visibile" dall'applicazione client realizzata con Visual Basic è, per definizione di OLE Server, "visibile" da tutti gli altri applicativi OLE Automation compatibili, ad esempio da Excel.

## Prima di cominciare, ancora due o tre premesse

I nostri esercizi, come detto, si dividono perlomeno in due tempi.

Nel primo tempo realizzeremo la parte server. Durante questo lavoro utilizziamo tecniche standard Visual Basic ed eseguiamo le operazioni necessarie alla creazione di un server (quelle appena elencate). Dobbiamo poi concretizzare, materializzare il server, ad esempio dobbiamo decidere se realizzare una libreria DLL o un eseguibile EXE.

Nel secondo tempo realizzeremo la parte client.

È chiaro che la messa a punto del complesso server/client richiederà, specie le prime volte, numerosi passaggi tra client e server. Inoltre la costruzione del server prevede una serie di passaggi obbligati. Se se ne salta uno, o se lo si sbaglia, il meccanismo non funziona ed è un po' difficile capire subito dove sta l'errore, specie se l'errore si verifica sul server quando lo si sta usando dal client. In questi casi, ovviamente, la messaggistica di errore è molto generica.

Cl.Calcolo(Q1,I1,P1)

Nella figura 4 vediamo come sfruttare la stessa libreria in una macro scritta con il Visual Basic di Excel 95.

Nella figura 3 vediamo una serie di box relative alla creazione e all'uso dell'OLE Server.

## Un OLE Server che esegue una Ricerca di Dati

Del secondo esercizio vediamo, in figura 5, subito la parte client.

C'è una form, con due "TextBox" e due "Command Button". Si scrive il nome di una città nella prima TextBox e si pigia il pulsante "Esegui". Questo, il cui codice è visibile in secondo piano, attiva una form, nella quale è riportato il nome della città. In questa form c'è un pulsante che serve per eseguire un'estrazione di dati, relativi alla città sopra indicata ed il conteggio dei record estratti. Il risultato di questo conteggio viene restituito all'applicazione chiamante.

Per richiamare una nuova istanza della classe:

```
Dim P2 As Prova2c
```

```
Set F2 = New Prova2c
```

per settare la proprietà X, che serve per passare al server il nome della città:

```
F2.X = T1
```

per eseguire il metodo SF, che esegue lo scorrimento dei dati:

```
F2.SF
```

per leggere la proprietà T, che riporta il valore del conteggio eseguito:

```
T2 = F2.T
```

Come si vede il meccanismo è semplicissimo.

Passiamo al server per vedere come va organizzato.

Il server dispone di una form, quella che abbiamo visto nella figura 5, e di un modulo di classe. Vediamo, nel collage di figura 6, i due gruppi di listati.

Nulla da dire su quello interno alla form del tutto analogo a quello da scrivere in un'applicazione normale.

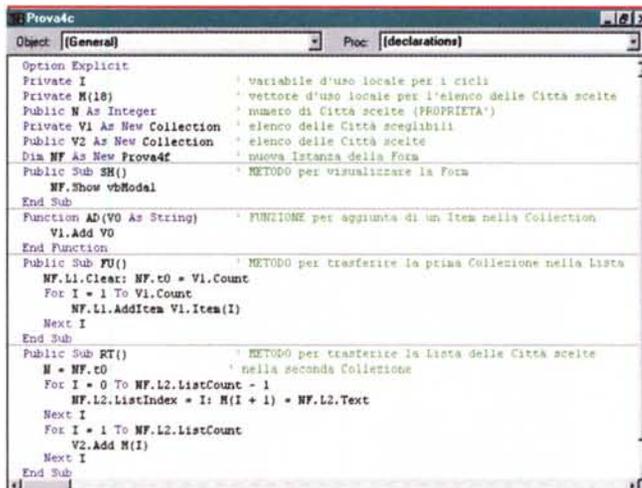
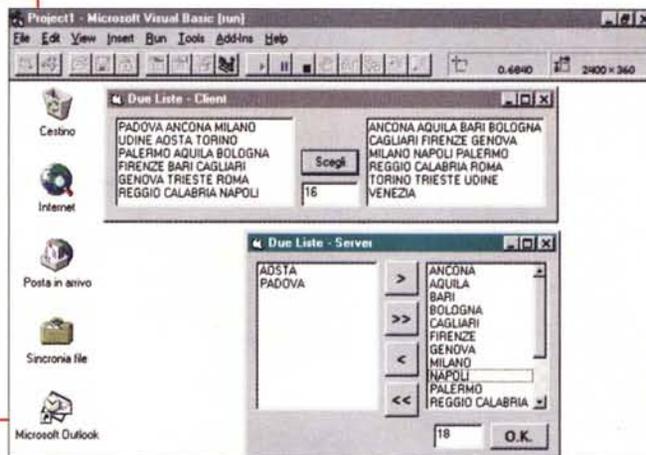
Per quanto riguarda il modulo di classe vediamo come vanno dichiarate le due variabili X e T, che costituiscono le nostre proprietà. E vediamo il metodo SF che si occupa di visualizzare la form:

```
Dim PF As New Prova2f
```

```
PF.T1 = X
```

```
Pf.Show vbModal
```

Figura 11 - Esercizio Prova4 - In alto il client ed in basso il server. Questa figura è quasi identica alla 7. In quel caso si trattava di un'unica applicazione VB. In questo caso, è il risultato della separazione, si tratta di un OLE Client che sfrutta un OLE Server. Le due figure non sono identiche in quanto nelle due form abbiamo inserito due TextBox in cui abbiamo riportato il numero degli elementi scelti.



scelta, il contenuto della seconda lista L2 nella seconda collezione V2. Abbiamo preferito spezzettare il processo per poterlo testare passo passo e quindi per poterlo capire meglio. Non mostriamo, in quanto sono identiche a quelle viste nell'esercizio Prova3, le routine interne alla form del server, quelle relative ai pulsanti per le scelte.

Figura 12 - Esercizio Prova4 - Listato del modulo di classe del server.

Il server dispone di ben quattro metodi, anzi di tre metodi e di una funzione. Il metodo SH mostra la form in modalità modale. La funzione AD serve per aggiungere un singolo elemento alla collezione V1, quella con l'elenco delle città da cui scegliere. Il metodo FU serve per trasferire la collezione V1, una volta completato il suo caricamento, nella prima lista L1. Il metodo RT è quello che trasferisce, una volta completato il lavoro di

A questo punto il controllo dell'applicazione è assunto dalla form, dalla quale si esce con il pulsante "Esci". Viene restituito il valore della proprietà T.

## Preparazione al quarto esercizio

Vogliamo proporvi un metodo di lavoro un po' "furbetto", utilizzabile nel caso in cui, ad esempio, il client disponga di una form sulla quale si eseguano una serie di impostazioni. Poi dal client si richiama il server, al quale invia dei dati. Il server fa le "sue cose" e poi restituisce altri dati al mittente, pardon al client.

Ebbene il metodo consiste nel costruire l'applicazione in modalità "normale", e poi, quando tutto funziona perfettamente, di spezzarla nei suoi due componenti, provvedendo alle necessarie correzioni.

In figura 7 vediamo in alto una form che mostra una lista di dati, una lista qualsiasi posta addirittura in una TextBox. Dalla form richiamiamo un'altra form che ha due ListBox, la prima delle quali viene riempita con la lista proveniente dalla prima form.

Compito della seconda form è quello di svolgere la funzione di "Finestra per Scelte Multiple". In pratica azionando i vari pulsanti, o facendo doppio click sugli elementi della lista, si scelgono una

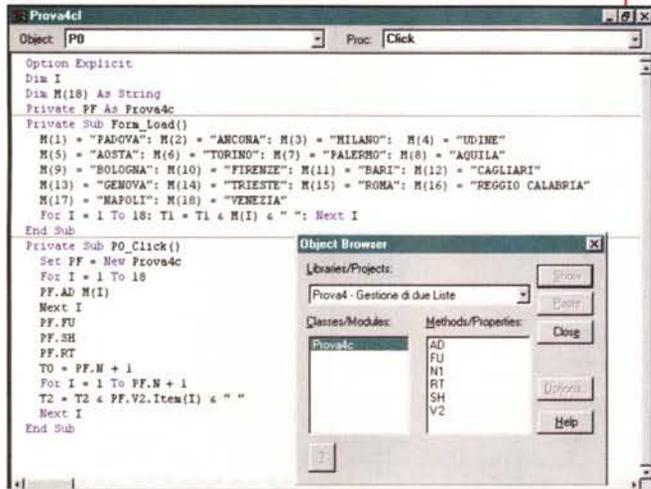


Figura 13 - Esercizio Prova4 - Listato del modulo della form del client. Compito del client è quello di mettere a disposizione una lista, cosa che facciamo in modo banale, e quello di richiamare i servizi del server, nel nostro caso di richiamare, uno dopo l'altro, i vari metodi del server. L'ordine di richiamo è obbligato. L'ultimo metodo, RT, restituisce la collezione V2, che contiene l'elenco delle città scelte. Questo viene posto nella seconda TextBox.

nella seconda delle due figure, serve per trasferire i dati delle celle di Excel in un vettore M() dal quale, poi, sfruttando il metodo AD, saranno spostate nella Collection V1 dell'OLE Server. I dati (l'elenco delle città scelte) ritorna, alla chiusura del server, nella Collection V2.

Nella successiva figura 11 vediamo come appaiono client e server una volta divisi. Rispetto alla versione "mono" abbiamo inserito un paio di TextBox che ci servono per visualizzare il numero degli Item scelti.

Nelle successive figure 12 e 13 vediamo il listato del modulo di classe, opportunamente commentato, in cui ci sono una proprietà, una funzione e tre metodi ed il listato di un eventuale client, che è analogo a quello visto in Excel. In questo caso buona parte del codice è "sprecata" per caricare in un modo banale il vettore M().

Non mostriamo i listati legati agli eventi sulla form, che sono del tutto identici a quelli visti nell'esercizio 3.

Sempre nella figura 13 vediamo come appare esauriente l'Object Browser che mostra la nostra libreria Prova4, la nostra classe Prova4c, con i suoi metodi e le sue proprietà.

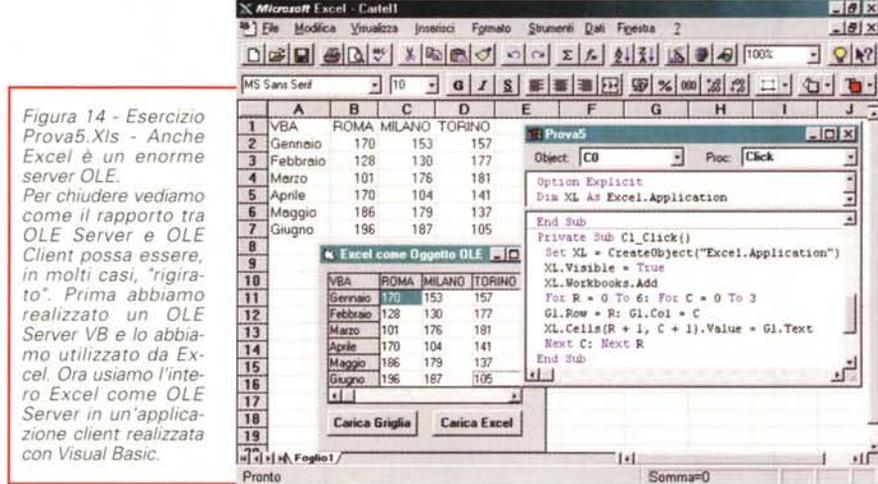


Figura 14 - Esercizio Prova5.xls - Anche Excel è un enorme server OLE. Per chiudere vediamo come il rapporto tra OLE Server e OLE Client possa essere, in molti casi, "rigirato". Prima abbiamo realizzato un OLE Server VB e lo abbiamo utilizzato da Excel. Ora usiamo l'intero Excel come OLE Server in un'applicazione client realizzata con Visual Basic.

## Conclusioni

Per chiudere, in figura 14, vi mostriamo un esercizio "contrario" rispetto ai precedenti. VB è il client che usa l'intero Excel come OLE Server.

Abbiamo raggiunto il nostro primo scopo, sappiamo "grosso modo" costruire un OLE Server.

Grosso modo perché l'OLE Server funziona, ma non è certamente ottimizzato. Non ci siamo preoccupati di valutare se ci conviene realizzare una DLL o un EXE. Non ci siamo occupati degli eventi di inizializzazione e di terminazione della Classe (a proposito, durante i primi esperimenti, i vostri server rimarranno spesso, e vostro malgrado, attivi: potete forzarne la chiusura premendo Ctrl+Alt+Canc e scegliendo "Termina Applicazione").

Ciò nonostante ci possiamo già rendere conto di quanto questo modo di lavorare per componenti possa costituire un modo per ottimizzare applicazioni complesse e per alleggerirne i costi di manutenzione

serie di elementi, che vengono riportati nella seconda ListBox. Finito di scegliere, klik su Ok, ritorno alla prima Box e riempimento della seconda TextBox della prima form.

La figura 7 mostra in maniera evidente i compiti delle due form. Nella successiva figura 8 vediamo i due listati, quello della seconda form, con tutti i suoi pulsanti, sullo sfondo, ed in primo piano quello della prima form.

Come si vede il grosso del lavoro lo svolge la seconda form. Inoltre il tipo di lavoro svolto da quest'ultima è assolutamente standardizzabile, in qualsiasi applicazione può servire una box con una doppia ListBox che serva per fare scelte multiple, per cui è opportuno realizzare un OLE Server che svolga tale servizio.

Un aspetto interessante di questo esercizio è l'aver utilizzato, per trasferi-

re famiglie di dati dall'applicazione client a quella server, e viceversa, delle "Collezioni", struttura, propria del Visual Basic, molto più versatile dei vettori, che tra l'altro non possono essere utilizzati come proprietà nei server.

## Conversione di un'applicazione normale in un OLE Server

Ed ecco "come viene" il quarto esercizio. In particolare vediamo come sia possibile, una volta realizzata l'applicazione DueListe come OLE Server, usarla anche da Excel (lo vediamo nelle due figure 9 e 10). Il grosso del listato VBA,

# FINALMENTE SPIEGATA LA FOTOGRAFIA DIGITALE

L'evoluzione digitale della fotografia fornirà ai fotografi professionisti e agli amatori appassionati di computer molte nuove opportunità.

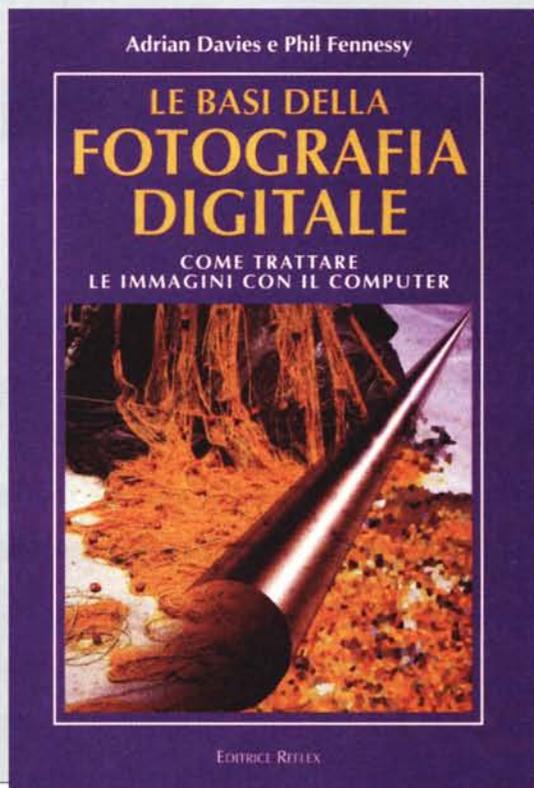
Questo libro spiega la tecnica degli strumenti (computer, scanner, stampa, trasmissione, ecc.) soffermandosi sul trattamento delle immagini. Ampio spazio è dedicato all'utilizzo degli strumenti di Photoshop ed al modo di avvicinarsi a questo eccellente programma di elaborazione dell'immagine.

Scritto da fotografi per i fotografi con un linguaggio semplice ed esauriente allo stesso tempo, il libro descrive con precisione e chiarezza tutto ciò che è necessario conoscere per orientarsi nel mondo della fotografia digitale. L'ampio glossario assicura che il significato di ogni termine sia perfettamente chiaro al lettore. Dopo, tutti i vostri dubbi sul fenomeno digitale saranno cancellati.

Dalla Editrice Reflex

**LE BASI DELLA FOTOGRAFIA DIGITALE**

L. 36.000, 136 pagine, 15x21cm.



Alcune tecniche spiegate nel libro.

**IN VENDITA NEI MIGLIORI NEGOZI DI FOTOGRAFIA ED IN LIBRERIA.**

POTETE RICHIEDERE IL VOLUME DIRETTAMENTE ALLA EDITRICE REFLEX. PAGAMENTO CON ASSEGNO BANCARIO, CARTA DI CREDITO (AMERICAN EXPRESS o CARTASI) OPPURE VERSANDO L'IMPORTO SUL CCP N. 82707001 INTESTATO A:

EDITRICE REFLEX, VIA DI VILLA SEVERINI 54, 00191 ROMA

TEL. 06-36308595 - 36301756 FAX 06-3295648