

- [Home Page](#) 
[Informazioni](#) 
[Aiuto](#) 

Corso base - Lezione 16

http://www.vbsimple.net/base/bas_16.htm

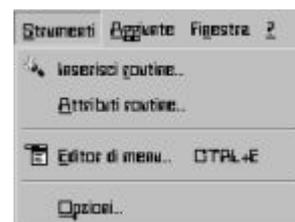
- [Alla scoperta dei menu di Visual Basic \(quinta parte\).](#)
- [Menu Strumenti.](#)

Il Corso Base si conclude con la visione di tutti i menu dell'IDE di Visual Basic. Alcune voci sono semplicissime da comprendere e presenti in quasi tutti i programmi per Windows, ma alcune sono realmente complesse e talvolta ostiche da utilizzare. Nelle lezioni 12-17 vedremo uno per uno tutti i menu ed approfondiremo le loro opzioni.

Menu Strumenti

Include alcuni comandi per svolgere operazioni di diverso genere mediante finestre di dialogo.

Le prime due operazioni consentono di lavorare con funzioni personalizzate nel nostro form, modulo  standard o di classe  o altro genere di pezzo di codice.



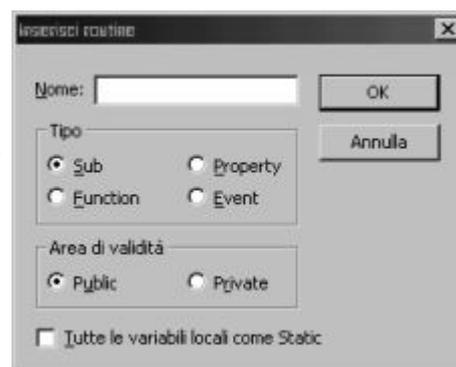
È presente anche un Editor di menu (visibile anche sulla Barra degli strumenti).

1. **Inserisci routine...**

Apre una finestra di dialogo per la creazione di una nuova routine.

Include una serie di pulsanti di opzione per decidere il tipo di routine da creare che può essere:

- **Sub** - Routine che non restituisce alcun valore in uscita (detta in altri linguaggi Procedura o funzione di tipo Void).
- **Function** - Routine che restituisce un dato in uscita.
- **Property** - [Proprietà](#)  di lettura e scrittura (Get e Let).
- **Event** - [Evento](#)  sempre pubblico; non esiste infatti un evento non pubblico.



È possibile anche decidere l'area di validità della routine da creare, che può essere:

- **Public** - Pubblica, cioè accessibile a tutte le parti del codice.
- **Private** - Privata, accessibile quindi soltanto alla sezione del codice in cui è inserita la funzione.

In ultimo è possibile anche definire se la routine sarà [Static](#) ovvero tutte le variabili

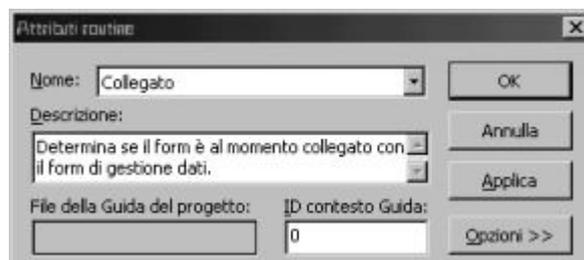
locali in essa contenute saranno di tipo Static ed il loro valore sarà mantenuto tra le varie chiamate alla routine.

Il comando Inserisci routine è uno dei comandi meno utilizzati dagli sviluppatori, sia per la relativa noia di utilizzo (anche se non è poi così complessa) sia perché è molto più rapido andare a scrivere la dichiarazione della funzione.

2. Attributi routine...

Mostra una finestra di dialogo - probabilmente la più complessa di Visual Basic - per la definizione degli attributi di una routine.

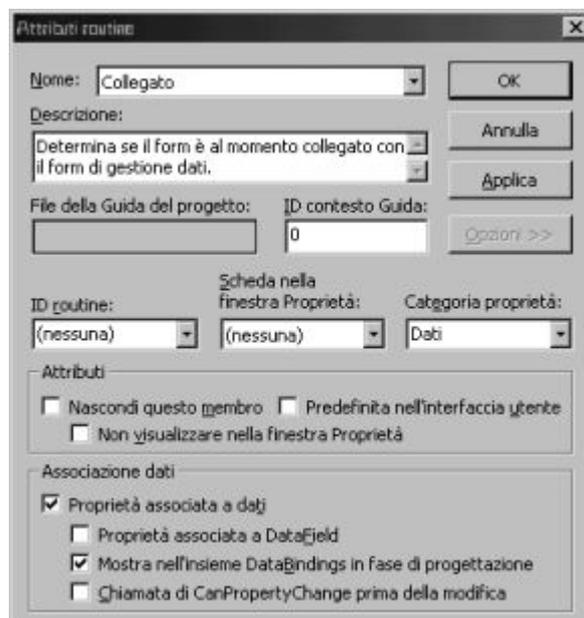
Esistono due modalità di visualizzazione di questa finestra di dialogo: quella ridotta (come quella qui a fianco) e quella dettagliata a cui si accede premendo il pulsante **Opzioni**. Sono in ogni caso mostrate soltanto le routine aggiunte e non quelle standard.



È presente il nome della routine di cui stiamo variandone gli attributi. Segue la descrizione della routine che come tale apparirà nel [Visualizzatore di oggetti](#)  quando lo sviluppatore andrà ad interrogare la routine in questione.

Il campo *ID contesto Guida* serve per associare una sezione della guida del progetto alla routine in questione. Se lo sviluppatore preme il tasto F1 sul nome della funzione nel Visualizzatore di oggetti apparirebbe la pagina dedicata.

Nella parte avanzata della finestra di dialogo sono presenti alcuni comandi particolari che consentono di estendere la funzionalità della funzione a migliorarne il funzionamento. Si raccomanda di non modificare tali valori fino a quando non si ha una discreta conoscenza del loro funzionamento.



In Visual Basic tutte le routine hanno un loro numero identificativo detto ID Routine o **Dispatch ID (DISPID)**. Il programmatore utilizza regolarmente il nome della routine da eseguire, ma in altri linguaggi quale il C++ è possibile richiamare la routine tramite il suo numero identificativo. La casella *ID Routine* può essere utilizzata per assegnare un DISPID alla routine. Ciò fa sì che la routine in questione sia legata alle azioni effettuate sul controllo. Ad esempio se assegnassimo ad una nostra routine il DISPID relativo alla proprietà Caption, ogni volta che la Caption del nostro controllo viene modificata la nostra routine viene eseguita automaticamente. I DISPID sono definiti come [costanti API](#) del protocollo [ActiveX](#).

La voce *Scheda nella finestra proprietà* serve per indicare quale pagina delle proprietà identifica la routine proprietà e sarà richiamata all'utilizzo del comando **Pagine proprietà**.

La *Categoria proprietà* serve per decidere a quale gruppo di proprietà la nostra routine sarà inserita.

La casella *Nascondi questo membro* fa sì che la routine sia invisibile nel **visualizzatore di oggetti** e nel sistema **Intellisense**. Il membro sarà comunque utilizzabile via codice.

La casella *Predefinita nell'interfaccia utente* imposta la nostra routine proprietà come proprietà predefinita; sarà pertanto possibile utilizzare il nome dell'oggetto per richiamare automaticamente la proprietà in questione.

L'attributo *Non visualizzare nella finestra Proprietà* nasconde la proprietà dalla finestra delle proprietà ma non dal visualizzatore di oggetti o nel sistema Intellisense.

Segue la sezione dedicata all'Associazione dati (binding); essa consente ad proprietà di essere direttamente collegata con i dati di un database, senza che sia richiesto un codice per la lettura e la scrittura del database. I dati associati sono gestiti automaticamente da VB ma affinché una proprietà sia associata ai dati è necessario scegliere una o due opzioni di binding.

La casella *Proprietà associata a DataField* collega i dati del campo indicato nella proprietà DataField del database associato alla proprietà in esame cui si riferisce questa casella.

La casella *Mostra nell'insieme DataBindings in fase di progettazione* aggiunge la proprietà all'insieme DataBindings in modo da poterla associare ad un campo specifico del database collegato.

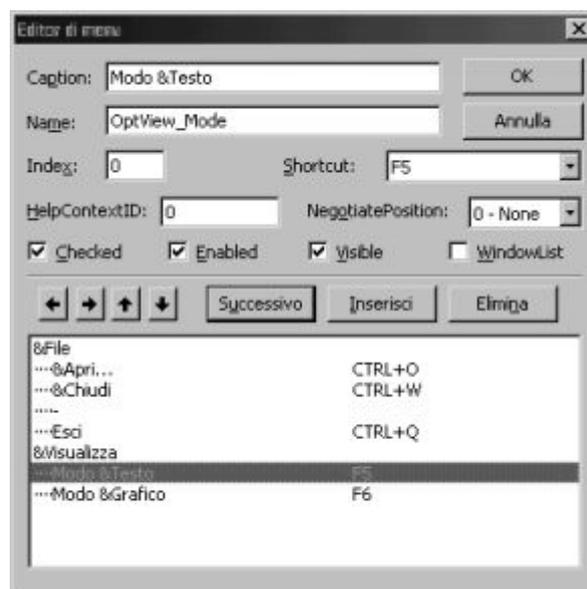
La casella *Chiamata di CanPropertyChange prima della modifica* effettua una chiamata alla funzione CanPropertyChange che si occuperà di controllare se è possibile cambiare i dati della proprietà (e quindi della tabella associata).

3. Editor di menu...

L'editor dei menu è uno strumento molto comodo per creare ed organizzare i menu di un progetto.

Nella parte superiore della finestra sono presenti i comandi per definire il testo del menu, il suo nome come oggetto, l'indice nel caso esso componga una matrice di menu con lo stesso nome. È possibile anche assegnare dei tasti scorciatoia (Shortcut) ad ogni voce del menu ed una pagina del sistema guida assegnato.

Interessante un breve discorso riguardante la Caption del menu: per assegnare un tasto di scelta rapida al menu o ad una sua voce far precedere la lettera da sottolineare come tasto di rapida con **&**.



Per creare invece una linea di separazione tra le varie voci la *Caption* deve essere un semplice trattino orizzontale, come mostrato nella figura subito sotto la voce **Chiudi**.

Un discorso separato lo merita la voce *NegotiatePosition*. In VB infatti è possibile incorporare degli ambienti di comunicazione [OLE](#). Ciò fa sì che l'ambiente stesso, con la sua barra di menu venga inserito all'interno del form. La barra dei menu del programma ospite prenderà il posto della barra del nostro progetto.

Tuttavia, in taluni casi è necessario che la nostra barra dei menu non sparisca. La voce *NegotiatePosition* permette di determinare se una voce di menu primaria con tutte le sue sottovoci rimanga presente anche quando l'oggetto incorporato viene attivato e, se attivato, in quale posizione debba stare (sinistra, centro, destra). Tale opzione può essere utilizzata soltanto con voci di menu primarie.

Seguono quattro piccole *CheckBox* per determinare lo stato della voce di menu attuale: *Checked* indica se debba apparire un segno di spunta accanto alla voce; può essere utilizzata per determinare lo stato di una voce di menu; *Enabled* determina se il menu e le sue sottovoci sono attivati e rispondono agli eventi; *Visible* indica se il menu è visibile; *WindowList* indica se il menu conterrà l'elenco di tutte le finestra MDI Child di un form [MDI](#)  primario.

Nella parte inferiore della finestra sono presenti alcuni pulsanti ed una lista di voci di menu. Il menu è organizzato a gerarchie; un menu primario potrà contenere tante voci al suo interno. Ogni voce si distingue dalla rientranza verso destra rispetto al menu primario. Nella schermata di esempio possiamo vedere due menu principali di nome *File* e *Visualizza* i quali hanno rispettivamente 4 e 2 voci. È possibile riconoscere la posizione gerarchica di ogni voce di menu in base alla sua rientranza rispetto ai membri precedenti.

Sono presenti anche 4 pulsanti a forma di freccia, ognuna indicante una direzione: la freccia verso **sinistra** serve per spostare una voce di menu verso sinistra facendola quindi salire di livello gerarchico; la freccia verso **destra** rientra il menu verso destra creando quindi una voce di secondo, terzo o quarto livello.

La freccetta verso l'**alto** serve semplicemente a spostare la voce di menu nella posizione precedente ed invertire con la voce che la precede. Viceversa la freccia verso il **basso** sposta una voce di menu in basso e porta la voce successiva al posto di quella che viene abbassata.

Il pulsante *Successivo* permette di spostare l'azione da una voce di menu alla successiva; il pulsante *Inserisci* consente di inserire una nuova voce di menu; il pulsante *Elimina*, invece elimina la voce corrente.

Questo strumento si presenta come uno dei più complessi ma il funzionamento è abbastanza istintivo e semplice. Probabilmente il miglior modo per impararne il funzionamento consiste proprio nel provare le varie funzioni in esso presenti.

4. Opzioni...

Permette la personalizzazione dell'ambiente di sviluppo di Visual Basic. Consente di attivare o disattivare le caratteristiche del codice, quali il sistema [Intellisense](#), il controllo sintattico durante la scrittura ed una serie di suggerimenti aggiuntivi.

Consente anche di impostare i colori della scrittura e vari stili; determinare se salvare automaticamente o su richiesta, il tipo di compilazione, etc...

Si tratta in generale di caratteristiche da impostare una volta ed in linea generale da non toccare più. Esse vengono automaticamente salvate e ripristinate al successivo riavvio di VB.

Questo menu, seppur piccolo, si presenta come il più complesso a causa delle numerose caratteristiche di ogni sua voce. In linea generale comunque è un menu che viene evitato se non per ottimizzare al meglio il funzionamento di un progetto.

[Fibia FBI](#)
19 Luglio 2001



[Torna alla quindicesima lezione](#)

[Vai alla diciassettesima lezione](#)

