




## Corso Intermedio - Lezione 2

[http://www.vbsimple.net/intermed/int\\_02.htm](http://www.vbsimple.net/intermed/int_02.htm)


- [Preparare il progetto.](#)
- [È ora di cominciare?.](#)
- [Le prime decisioni.](#)

Durante tutto il corso intermedio ci occuperemo di sviluppare passo dopo passo un progetto di un software per l'archiviazione dei CD musicali. Servirà per imparare con la pratica l'utilizzo dei controlli e delle prime istruzioni di Visual Basic.

Prima di iniziare a disporre controlli sulla superficie del primo form e scrivere il relativo codice ci occuperemo di impostare le caratteristiche di base del form.

Inizialmente non faremo uso di controlli esterni anzi sfrutteremo i controlli intrinseci ([Thunder](#)) di Visual Basic ma, prima ancora di occuparci di essi, regoleremo alcune proprietà  del form:

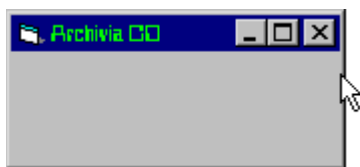
● **Name**: assegna un nome che identifica **univocamente** il nostro form; non sarà tuttavia possibile modificare questo valore durante la fase di esecuzione; questa infatti è una proprietà in lettura e scrittura durante la [fase di progettazione](#) ma di sola lettura (quindi non modificabile) durante la [fase di esecuzione](#).

● **BorderStyle**: imposta lo stile del bordo del form ed il suo comportamento. Ammette i valori numerici compresi tra 0 e 5. Essa determina la presenza della barra del titolo e l'eventuale numero di [pulsanti su di essa](#) e contiene un valore dell'[enumerazione](#)  [FormBorderStyleConstants](#). È una proprietà di sola lettura durante la fase di esecuzione.

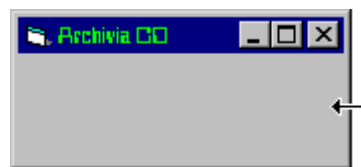
Nella maggioranza dei casi il valore da assegnare a questa proprietà sarà

**1 = vbFixedSingle** se il nostro form non dovrà essere ridimensionabile oppure

**2 = vbSizable** se esso dovrà essere ridimensionabile. L'impostazione di uno di questi due valori determinerà infatti il cambiamento automatico del puntatore del mouse quando esso si posa su uno dei bordi del form, in modo da permetterne il ridimensionamento tramite mouse.






**Figura 1**  
(BorderStyle = 1)




**Figura 2**  
(BorderStyle = 2)


L'impostazione della proprietà BorderStyle sul valore **2 = vbSizable** determina anche la presenza **iniziale** dei pulsanti sulla barra del titolo. Iniziale perché se impostassimo questa

proprietà ai valori 1 o 2 sarà comunque possibile regolare la presenza dei pulsanti di riduzione ad icona  e di ingrandimento  o ripristino  mediante le proprietà **MaxButton** e **MinButton**. Anche quest'ultime sono proprietà in lettura e scrittura durante la fase di progettazione e di sola lettura durante la fase di esecuzione.

L'impostazione al valore 2 permette all'utente il ridimensionamento del form ma esso sarebbe inutile se il programmatore non provvedesse una gestione automatica dei controlli sul form in modo da ingrandirli o ridurli in funzione dello spazio libero.

Senza complicarci troppo la vita impostiamo tale valore a **1 = vbFixedSingle** e ripristiniamo il pulsante di riduzione ad icona  tramite la proprietà **MinButton** impostata a **True**.


● **Caption**: quasi superflua la descrizione... determina il testo presente sulla barra del titolo del nostro form (nel nostro caso "Archivia CD").

● **ControlBox**: consente di far sparire tutti i pulsanti e la casella di controllo  del form. Se tale proprietà è impostata su **False** il valore delle proprietà **MinButton** e **MaxButton** è ignorato. La proprietà è modificabile soltanto in fase di progettazione. Nel nostro esempio lasceremo il valore di default **True**.

● **Font**: come dice il nome stesso consente di modificare il font (carattere, grandezza e stile) delle operazioni di scrittura e dei controlli che verranno aggiunti in seguito all'impostazione di questa proprietà.

Per operazioni di scrittura si intende l'utilizzo dell'istruzione **Print** per scrivere direttamente sulla superficie del form e non all'interno di qualche controllo.

Il vantaggio di impostare questa proprietà subito sta nel fatto che tutti i controlli che verranno inseriti sul form successivamente ereditano il valore di questa proprietà. Personalmente detesto il carattere di default *MS Sans Serif* e tendo ad utilizzare quasi sempre il più moderno *Tahoma*. Impostando sin da subito il carattere *Tahoma* tutti i controlli che aggiungeremo in seguito avranno come font di default il *Tahoma* scelto in precedenza. Si tratta ovviamente di scelte personali, ma anche per questa volta seguirò il mio **personalissimo** gusto utilizzando appunto *Tahoma*.

● **Icon**: l'icona che apparirà sulla casella di controllo del nostro form, nell'elenco delle applicazioni attive (*tasti ALT+TAB*) ed in tanti altri casi al posto dell'icona di VB  standard. Le icone possono contenere al loro interno una o più immagini in funzione della grandezza richiesta. Ad esempio l'icona sulla casella di controllo del form è generalmente grande 16x16 pixel; se all'interno del file .ICO è presente un'immagine di tale risoluzione essa sarà utilizzata altrimenti ne verrà scelta un'altra ed adattata mediante ingrandimento o riduzione della stessa.

Nel nostro esempio utilizzeremo un'icona a forma di CD in due risoluzioni: 16x16 pixel la prima e 32x32 pixel la seconda. Le due immagini sono molto simili; la scelta di due icone sta nel fatto che se avessimo utilizzato un'icona troppo piccola oppure una troppo grande al momento della visualizzazione la resa grafica sarebbe stata davvero scadente... **provare per credere**.

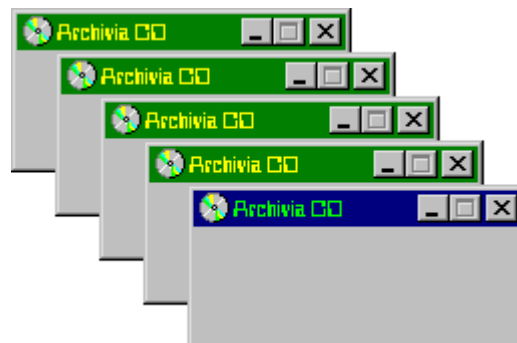
L'icona scelta è scaricabile cliccando sull'icona a fianco! 

Le icone possono essere create tramite il programma Microsoft Image Editor © (IMAGEDIT.EXE) presente nel CD di Visual Basic.

● **ShowInTaskBar**: determina se il form in esecuzione sarà visibile sulla barra delle applicazioni. La sua utilità si rivela in quei casi in cui un progetto è composto da due o più forms che sono contemporaneamente caricati e mostrati all'utente. Non è molto gradevole riempire la barra delle applicazioni con 5 o 6 icone rappresentative dello stesso programma; la proprietà *ShowInTaskBar* del form principale sarà impostata su **True** mentre quella degli altri form sarà impostata su **False**.

La regola non è assoluta anzi soggetta ad una fondamentale eccezione!

Se i forms (uguali o differenti, non importa) caricati hanno la proprietà *ShowInTaskBar* impostata su **True** ma sono mostrati in modalità modale la regola precedente non viene applicata. Sarà mostrata sulla barra sempre una sola icona ovvero quella del primo form caricato e mostrato sullo schermo.

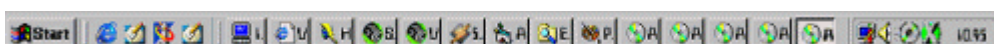


Viceversa se i forms sono mostrati in modalità non modale appariranno sulla barra delle applicazioni più icone ognuna delle quali rappresentativa del suo form.



**Figura 4**

*(I forms sono visualizzati in modalità modale)*



**Figura 5**

*(I forms sono visualizzati in modalità non modale)*

Nelle due figure la barra delle applicazioni rappresenta lo stesso numero di applicazioni aperte ma mentre nella prima figura il nostro programma (quello con l'icona a forma di CD) occupa solo una posizione con una sola icona, nel secondo caso le icone sono addirittura 5, una per ogni form mostrato in modalità non modale.

Nella maggioranza dei casi la visualizzazione di un form secondario viene fatta in modalità modale ovvero bloccante il form chiamante e pertanto non sarà necessario preoccuparsi di disabilitare la proprietà *ShowInTaskBar*.

Viceversa è importante che il form principale sia visibile sulla barra delle applicazioni e pertanto il valore della sua proprietà *ShowInTaskBar* sia **True**, per permettere all'utente il cambio della finestra attiva ed il successivo ritorno cliccando semplicemente l'icona sulla barra delle applicazioni.


● **StartPosition**: determina la posizione iniziale del form ed utilizza i valori

dell'[enumerazione](#)  [StartUpPositionConstants](#). Il significato dei suoi valori è:

- 0 = Manual (***vbStartUpManual***)  
La posizione del form è data esclusivamente dalle proprietà Left e Top.
- 1 = CenterOwner (***vbStartUpOwner***)  
La posizione del form corrisponde al centro del form che l'ha richiamato. Quando ad esempio il form principale si trova in uno degli angoli dello schermo e viene richiesta la visualizzazione di un form secondario con proprietà *StartUpPosition* impostata su questo valore, il form secondario apparirà al centro del form primario che l'ha mostrato ovvero al centro dell'angolo in cui si trova il form principale. Se il form cui si riferisce questa proprietà è quello principale il comportamento sarà lo stesso del valore 2 = ***CenterScreen*** ovvero esso apparirà al centro dello schermo.
- 2 = CenterScreen (***vbStartUpScreen***)  
La posizione iniziale del form è il centro dello schermo e sono ignorate le proprietà di posizione Left e Top.
- 3 = Windows Default (***vbStartUpWindowsDefault***)  
La posizione è stabilita automaticamente da Windows e le proprietà Left e Top sono ignorate. I forms alla figura 3 utilizzavano questo valore; Windows infatti si occupa di disporre le finestre a cascata.



La proprietà *StartUpPosition* regola soltanto la posizione **iniziale** del form. Non è possibile, ad esempio, utilizzarla per centrare il form dopo la sua apparizione. Per centrare un form via codice utilizzare l'esempio proposto nell'[HowTo dedicato](#).

● **WindowState**: determina lo stato della finestra ed utilizza i valori dell'enumerazione  [FormWindowStateConstants](#). Determina ovvero se il form è nello stato normale, ridotto ad icona oppure massimizzato; corrisponde in sintesi agli stati definibili tramite i pulsanti di riduzione, ingrandimento e ripristino ed è utilizzabile anche quando il form è definito a grandezza fissa, senza i pulsanti di ingrandimento o di riduzione ad icona.

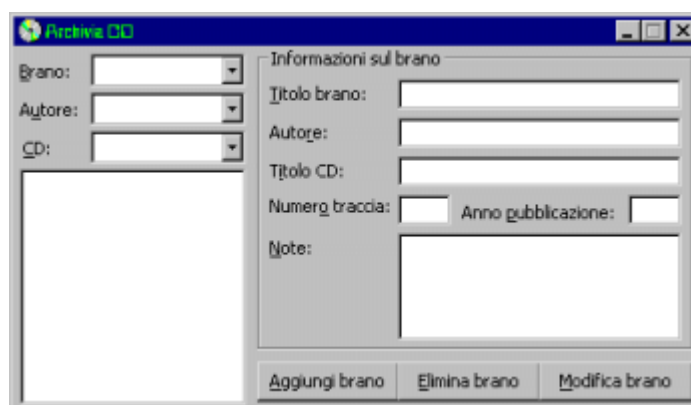
Viene utilizzata per modificare lo stato della finestra ad esempio richiamare la riduzione ad icona durante certe circostanze.

In quei form a grandezza variabile (vedi la proprietà ***BorderStyle***) prima di ricalcolare la posizione che dovranno assumere i controlli è necessario verificare se la finestra non sia ridotta ad icona tramite la lettura di questa proprietà.

Altresì non è possibile modificare la grandezza e la posizione di un form massimizzato o ridotto ad icona, pertanto prima di effettuare ogni tentativo di spostamento o modifica della grandezza è necessario verificare che il valore di questa proprietà sia quello dello stato normale (***vbNormal***).

---

Vediamo brevemente come apparirà la nostra interfaccia utente al fine di semplificare la comprensione del problema.

**Figura 6**

Il form è diviso in due sezioni verticali: quella a sinistra consente di effettuare le ricerche dei brani mediante un semplice meccanismo di filtro per nome del brano, nome dell'autore o titolo del CD.

La sezione a destra, invece, contiene i dati informativi sul brano scelto. Mediante i pulsanti che si trovano sotto di questa sarà possibile aggiungere un brano all'archivio, eliminare quello appena scelto o modificarne i dati.

Questo è ciò che proveremo a sviluppare nelle prossime lezioni, ma prima di iniziare ad implementarlo è fondamentale affrontare alcune scelte che determineranno il proseguimento dello sviluppo. In realtà non siamo ancora in grado di scegliere con precisione la soluzione migliore; pertanto ci limiteremo ad analizzare le soluzioni più semplici e scegliere quella che sembra offrire le caratteristiche migliori.

La prima decisione riguarda naturalmente l'archivio!

Che tipo di archivio scegliere? Ne abbiamo a disposizione una gamma davvero ampia... quale sarà il migliore per il nostro progetto? Un file di testo? Un file binario? Un database? Un foglio di calcolo?

...

...

...

**NO! NO! NO! e NO!**

Questa metodologia di risoluzione dei problemi è nettamente sbagliata e superficiale. È errore molto comune mettere subito le *mani in pasta* nel progetto.

Prima di poter effettuare una scelta tra la gamma di soluzioni disponibili è fondamentale capire COSA ci serve effettivamente.

Ci serve un archivio? Beh, tutti quelli accennati precedentemente lo sono.

Non ci serve soltanto un archivio, un contenitore per i nostri dati. Ci serve piuttosto una soluzione per il nostro problema, problema quello dell'aggiungere, ricercare, modificare ed eliminare dati da un archivio.

Abbiamo già deciso (vedi lezione precedente) che il nostro programma sarà utilizzato al massimo da un solo utente per volta (*monoutente*), ovviamente non sarà connesso in Internet e conterrà pochi dati ovvero non più di 10.000 brani.

Soltanto **dopo** aver fatto queste scelte potremo iniziare a porci il problema della scelta dell'archivio, scelta che verrà fatta **in funzione** delle decisioni prese e mai viceversa.

Essendo i requisiti del programma molto esigui sarà facile effettuare la scelta dell'archivio da utilizzare. Propenderemo per un semplice file binario con campi di larghezza fissa; sebbene l'utilizzo di un file di testo semplice sarebbe stato sicuramente più semplice le operazioni di ricerca dei dati sarebbero state più complesse e l'utilizzo delle note (vedi figura 6) avrebbe complicato di molto le cose.

Tutti gli altri archivi - database Access, SQL o cartella di Excel - possiedono tutte le caratteristiche per essere utilizzati in questo nostro progetto e sicuramente qualcosa in più ma per essere utilizzati richiedono una maggiore quantità di risorse in termini di:

- conoscenze;
- memoria;
- spazio su disco;
- difficoltà di installazione.

Nella prossima lezione procederemo con lo sviluppo del progetto fin qui analizzato utilizzando come archivio un file binario con record a lunghezza fissa.

[Fibia FBI](#)

29 Dicembre 2001



[Torna alla prima lezione](#)

[Vai alla terza lezione](#)

