



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

## Il piano Zero OCX

[http://www.vbsimple.net/news/news\\_12.htm](http://www.vbsimple.net/news/news_12.htm)

Programmo in Visual Basic® dal 1998 avendo iniziato con il *VB5 Control Creation Edition*, un'edizione gratuita di VB limitata alla sola creazione di controlli utente e alla loro compilazione in [OCX](#).

Sembra quasi un controsenso ma da un paio di anni conduco una personale battaglia contro quella dilagante moda, seguita da molti *programmatori* VB, che propone gli OCX come la soluzione alla maggior parte dei problemi.

Per un programmatore alle prime armi l'aggiunta di un componente esterno all'interno di un proprio progetto è come una manna dal cielo: una parte, spesso difficile, del lavoro da svolgere è già stata sviluppata da qualcun altro e la sua fruizione richiede un semplice doppio click e poche righe di codice.

In realtà l'aggiunta di OCX esterni è molto spesso causa di problemi ed i punti a sfavore valgono quasi sempre ben più dei punti a favore.

Un OCX è un controllo utente, più o meno semplice, compilato in codice binario da collegare ad un altro progetto e può velocizzare di molto lo sviluppo di un progetto in corso, ma non è tutto oro quello che luccica:

- **Ogni OCX deve essere registrato!**

Tutti gli OCX sono componenti [ActiveX](#)® e come tali devono necessariamente essere registrati nel sistema. La registrazione dei componenti viene quasi sempre effettuata in maniera invisibile dal programma di installazione dell'applicativo. Registrare un componente significa scrivere ProgID, [CLSID](#) e dati informativi sulla [TypeLibrary](#) nel registro di Windows.

Dov'è il problema?

La dilagante moda degli OCX è una delle principali ragioni del progressivo rallentamento dei nostri computer nonostante la sempre più elevata velocità di elaborazione dei processori.

All'installazione di Windows il [registro di sistema](#) occupa dai 5 ai 10 MB a seconda della configurazione del PC. Dopo circa un anno il registro di un utente medio può persino quadruplicare di dimensione.

Purtroppo non è solo un problema di spazio occupato su disco rigido!

Per sua natura Windows tende a tenere tutto il file di registro in memoria e ciò può significare utilizzare un computer con svariati megabyte di memoria occupati dai dati di configurazione.

- **Gli OCX hanno files di dipendenza!**

Una discreta parte degli OCX è scritta in VB e come tali pertanto richiedono il set minimo di librerie di runtime installate sul computer in cui il componente viene

eseguito. Ciò significa che un progetto scritto in VB6 può richiedere l'installazione delle librerie di runtime di Visual Basic 4, 5 e 6 in funzione della versione dell'ambiente di sviluppo con cui ogni componente è scritto.

Molto spesso manca un elenco dei files da cui l'OCX dipende e ciò comporta la difficile installazione e quasi sempre il mancato funzionamento dell'applicazione che ne usufruisce su un computer diverso da quello in cui è stata sviluppata. In alcuni casi l'installazione di un OCX può provocare il blocco totale della macchina!

A questo va ad aggiungersi il fatto che un simbolico programma scritto in VB6 richiede non solo le librerie di runtime di VB6, ma anche quelle necessarie al corretto funzionamento dell'OCX (ovvero quelle di VB4 o VB5). I requisiti di memoria quindi possono facilmente raddoppiare rispetto le reali necessità del programma.

- **Molto spesso gli OCX sono coperti da copyright!**

Tantissimi OCX vengono rilasciati coperti da copyright e ce li si ritrova installati nel computer senza conoscerne le fonti. L'utilizzo di un componente di terze parti coperto da copyright senza una concessione del suo autore costituisce un vero e proprio reato.

- **Gli OCX possono nascondere virus ed attivarsi facilmente!**

Quasi tutti sanno che i files .EXE possono essere infetti o possono danneggiare il computer in cui sono eseguiti, ma quanti sanno che anche gli OCX sono esposti allo stesso rischio?

Un OCX infetto anzi ha più possibilità di infezione di un EXE. L'unico modo di lanciare un programma .EXE infatti è quello di cliccare due volte sull'icona del programma oppure di scrivere il nome sulla riga di comando.

Un OCX, invece, viene attivato ogni volta che un programma ne richiede l'automazione; un componente infetto viene pertanto attivato ogni volta che il nostro applicativo ne istanzia una copia sul form ma non solo! Lo stesso ambiente di VB può attivare il virus; tutti gli ActiveX possiedono 3 funzioni standard che vengono richiamate ogni volta che viene richiesta la [registrazione del componente](#).

Ecco che il virus può essere attivato anche senza lanciare il programma che usufruisce dell'OCX. Lo stesso pacchetto di installazione di un programma richiama le funzioni di registrazione del componente e con esse darà il via all'infezione del computer.

- **Gli OCX sono compilati!**

Escludendo quei rarissimi casi in cui il codice sorgente accompagna l'OCX in linea generale gli OCX sono dei programmi veri e propri e come tali possono essere nocivi in mille maniere.

Inoltre, non essendo correlati di codice sorgente non possono essere personalizzati o modificati a seconda delle esigenze influenzando quindi pesantemente le scelte implementative di un progetto.

Sono naturalmente presentati solo gli aspetti negativi derivanti dall'uso degli OCX in ambiente VB e talvolta è necessario affidare certe funzioni a controlli compilati perché scriverli da capo richiederebbe tanto tempo aggiuntivo, senza fornire le stesse prestazioni.

Un esempio lampante sono i *Microsoft Windows Common Controls*, una serie di controlli standard in Windows, alcuni dei quali davvero complessi (ad esempio la *TreeView*) e pensare di dover fare a meno di ogni OCX è un'idea tanto stupida quanto sviluppare un progetto fatto da soli OCX.

Mi sono invece sempre opposto e mi oppongo all'uso indiscriminato di OCX per le più stupide funzioni quando basterebbero pochissime righe di codice o un più semplice modulo già pronto da inglobare all'interno del progetto.

Controlli quali la *Microsoft Common Dialog* richiedono pochissime righe di codice per svolgere le stesse funzioni ed addirittura superare i limiti del controllo già pronto. Inoltre trovo assolutamente insensato includere i Common Controls solo per utilizzare controlli banali come la *Progress Bar*, davvero semplice da riprodurre (vedi ad esempio i controlli [FBI Shape Progress Bar](#) ed [FBI Graph Progress Bar](#)) o la Status Bar, che può essere simulata con una banale Label. Voglio anche ricordare che un controllo *Tabbed Dialog* a schede è già incluso all'interno dei *Common Controls* e quindi considero sciocco utilizzare contemporaneamente i suddetti OCX.

Vedo in giro progetti che fanno uso di OCX per la creazione di MessageBox, di anteprima di stampa oppure di report dove effettivamente non servono. Escludendo i controlli complessi (griglie o strutture), nel 90% dei casi è possibile sviluppare o trovare gratuitamente una classe o un modulo in grado di svolgere le stesse funzioni. Ce ne guadagnerà il nostro programma in velocità e requisiti e sicuramente l'installazione del prodotto sarà molto più semplice.

Per queste ragioni al giorno d'oggi all'interno di VBSimple non è presente nessun OCX ma soltanto un paio di [controlli utente](#) sviluppati in VB, non compilati, da includere per intero all'interno del progetto da sviluppare.

[Fibia FBI](#)

9 Novembre 2002



[Torna all'indice degli Articoli](#)

---