

# Chương Một - Hoan nghênh đến với VB6

## Chào mừng bạn đến với Visual Basic 6

Dùng VB6 là cách nhanh và tốt nhất để lập trình cho Microsoft Windows. Cho dù bạn là chuyên nghiệp hay mới mẻ đối với chương trình Windows, VB6 sẽ cung cấp cho bạn một bộ công cụ hoàn chỉnh để đơn giản hóa việc triển khai lập trình ứng dụng cho MSWindows.

**Visual Basic là gì?** Phần "**Visual**" đề cập đến phương pháp được sử dụng để tạo giao diện đồ họa người dùng (Graphical User Interface hay viết tắt là GUI) . Có sẵn những bộ phận hình ảnh, gọi là controls, bạn tha hồ sắp đặt vị trí và quyết định các đặc tính của chúng trên một khung màn hình, gọi là form. Nếu bạn đã từng sử dụng chương trình vẽ chẳng hạn như Paint, bạn đã có sẵn các kỹ năng cần thiết để tạo một GUI cho VB6.

Phần "**Basic**" đề cập đến ngôn ngữ BASIC (Beginners All-Purpose Symbolic Instruction Code), một ngôn ngữ lập trình đơn giản, dễ học, được chế ra cho các khoa học gia (những người không có thì giờ để học lập trình điện toán) dùng.

Visual Basic đã được ra từ MSBasic, do Bill Gates viết từ thời dùng cho máy tính 8 bits 8080 hay Z80. Hiện nay nó chứa đến hàng trăm câu lệnh (commands), hàm (functions) và từ khóa (keywords). Rất nhiều commands, functions liên hệ trực tiếp đến MSWindows GUI. Những người mới bắt đầu có thể viết chương trình bằng cách học chỉ một vài commands, functions và keywords. Khả năng của ngôn ngữ này cho phép những người chuyên nghiệp hoàn thành bất kỳ điều gì nhờ sử dụng ngôn ngữ lập trình MSWindows nào khác.

Người mang lại phần "Visual" cho VB là ông Alan Cooper. Ông đã gói môi trường hoạt động của Basic trong một phạm vi dễ hiểu, dễ dùng, không cần phải chú ý đến sự tinh xảo của MSWindows, nhưng vẫn dùng các chức năng của MSWindows một cách hiệu quả. Do đó, nhiều người xem ông Alan Cooper là cha già của Visual Basic.

Visual Basic còn có hai dạng khác: **Visual Basic for Application (VBA)** và **VBScript**. VBA là ngôn ngữ nằm phía sau các chương trình Word, Excel, MSAccess, MSProject, .v.v.. còn gọi là Macros. Dùng VBA trong MSOffice, ta có thể làm tăng chức năng bằng cách tự động hóa các chương trình. VBScript được dùng cho Internet và chính Operating System.

Dù cho mục đích của bạn là tạo một tiện ích nhỏ cho riêng bạn, trong một nhóm làm việc của bạn, trong một công ty lớn, hay cần phân bố chương trình ứng dụng rộng rãi trên thế giới qua Internet, VB6 cũng sẽ có các công cụ lập trình mà bạn cần thiết.

## Các ấn bản Visual Basic 6

Có ba ấn bản VB6: Learning, Professional và Enterprise. Chúng ta hãy gát qua ấn bản Learning. Bạn có thể dùng ấn bản Professional hay Enterprise.

Ấn bản Professional cung cấp đầy đủ những gì bạn cần để học và triển khai một chương trình VB6, nhất là các control ActiveX, những bộ phận lập trình tiền chế và rất hữu dụng cho các chương trình ứng dụng (application programs) của bạn trong tương lai. Ngoài đĩa compact chính cho VB6, tài liệu đính kèm gồm có sách Visual Studio Professional Features và hai đĩa CD Microsoft Developer Network (MSDN).

Ấn bản Enterprise là ấn bản Professional cộng thêm các công cụ Back Office chẳng hạn như SQL Server, Microsoft Transaction Server, Internet Information Server.

## Cài đặt VB6

Để cài đặt VB6, máy tính của bạn cần phải có một ổ đĩa CD-ROM (CD drive) . Bạn cần ít nhất 32 MB RAM, 2 GB hard disk và CPU Pentium II. Khi bỏ VB6 CD vào CD drive, nó sẽ tự khởi động để display menu cho bạn chọn những thứ gì cần Setup, hãy click **Install Visual Basic 6.0** để cài VB6.

Ngoại trừ các file hệ điều hành (Operating System) trong thư mục (folder) \Os, các file trong đĩa compact đều không bị nén. Vì thế, bạn có thể sử dụng chúng trực tiếp từ đĩa. Ví dụ, có nhiều công cụ và thành phần trong folder \Tools vốn có thể được cài đặt trực tiếp từ CD-ROM. Ngoài ra, bạn có thể chạy Setup khi nào cần thiết. Ví dụ, bạn có thể chạy Setup để cài đặt lại Visual Basic trong folder khác, hoặc để cài đặt thêm bớt các phần của VB6. Nếu vì lý do gì hệ thống không install các đĩa compact MSDN (bạn sẽ khám phá ra điều này khi thấy Help không có mặt lúc chạy VB6), bạn có thể cài đặt chúng trực tiếp từ đĩa số 1 của bộ MSDN.

### Để bổ xung và xóa các thành phần VB:

1. Bỏ đĩa compact vào CD drive.
2. Nếu menu không tự động hiện lên thì chạy chương trình Setup có sẵn trong folder gốc trên đĩa compact.
3. Chọn nút **Custom** trong hộp thoại (dialog) **Microsoft Visual Basic 6.0 Setup**.
4. Chọn hay xóa các thành phần bằng cách check hay uncheck các hộp danh sách **Options** của dialog **Custom**.
5. Thực hiện các chỉ dẫn Setup trên màn hình.

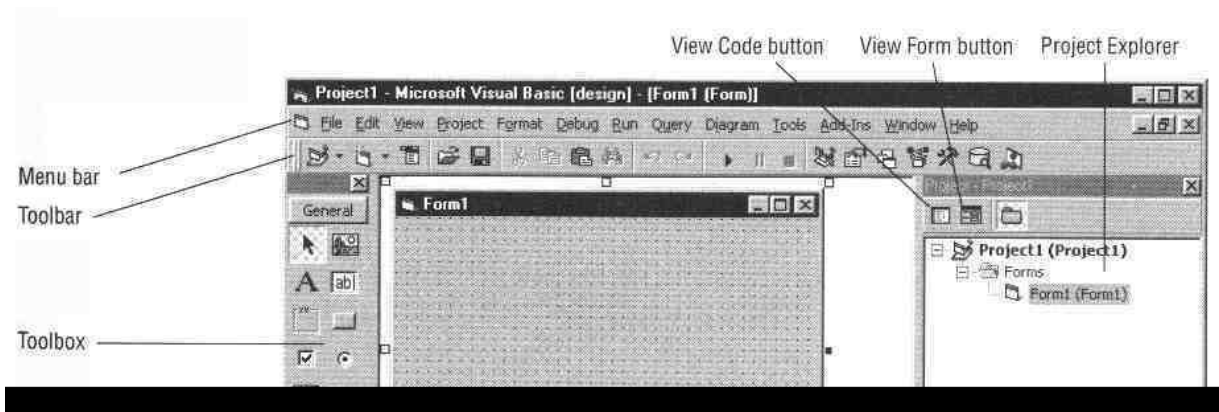
*Ghi chú: Trong lúc cài VB6, nhớ chọn Graphics nếu không bạn sẽ thiếu một số hình ảnh như icons, bitmaps v.v... Đáng lẽ Microsoft cho tự động cài đặt Graphics, tức là Default (không có nói gì) thì cài đặt Graphics.*

### Integrated Development Environment (IDE) của VB6

Khi khởi động VB6 bạn sẽ thấy mở ra nhiều cửa sổ (windows), scrollbars, v.v.. và nằm chồng lên là **New Project** dialog. Ở đây VB6 cho bạn chọn một trong nhiều loại công trình.



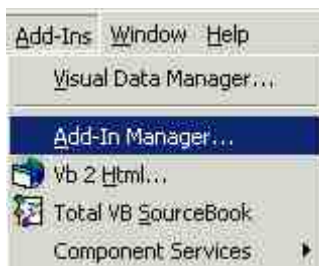
Chọn Standard EXE. Một lát sau trên màn ảnh sẽ hiện ra giao diện của môi trường phát triển tích hợp (Integrated Development Environment - IDE ) giống như dưới đây:



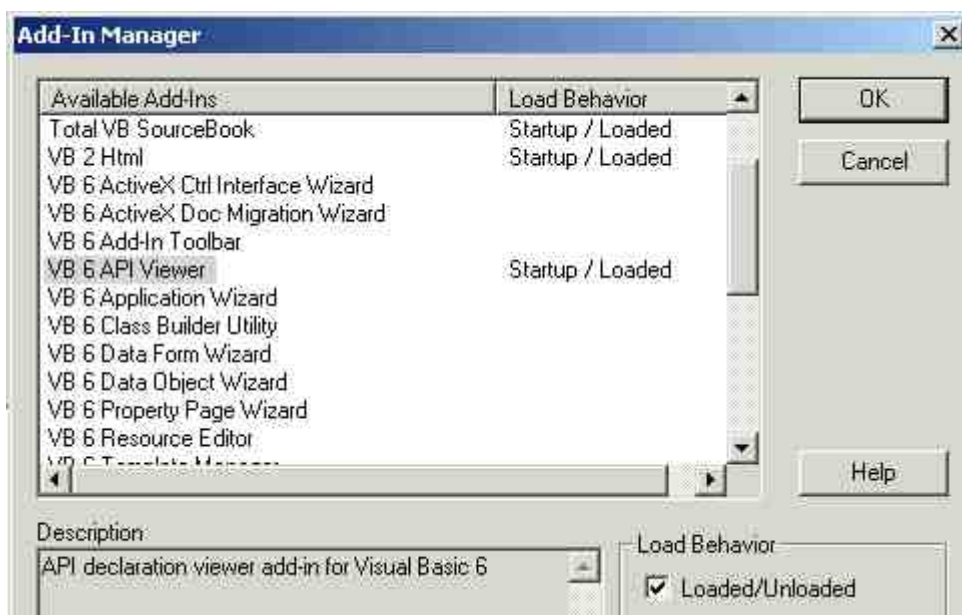
IDE của VB6 bao gồm các yếu tố sau:

## Menu Bar

Chứa đầy đủ các commands mà bạn sử dụng để làm việc với VB6, kể cả các menu để truy cập các chức năng đặc biệt dành cho việc lập trình chẳng hạn như Project, Format, hoặc Debug. Trong Menu Add-Ins có Add-Ins Manager cho phép bạn gắn thêm những menu con nhiệm ý để chạy các chương trình lợi ích cho việc lập trình.



Trong Add-Ins Manager dialog bạn chọn một Add-In rồi check một hay nhiều hộp trong khung Load behavior:



## Toolbars (Debug, Edit, form Editor, Standard)

Các toolbars có hình các icons cho phép bạn click để thực hiện công việc tương đương với dùng một menu command, nhưng nhanh và tiện hơn. Bạn dùng menu command **View | Toolbars** (click lên

menu command View cho popupmenu hiện ra rồi click command con Toolbars) để làm cho các toolbars hiện ra hay biến mất đi. Bạn có thể thay đổi vị trí một toolbar bằng cách nắm vào hai gạch vertical nằm bên trái toolbar rồi dời toolbar đi chỗ khác (nắm ở đây nghĩa là để pointer của mouse lên chỗ chấm đỏ trong hình phía dưới rồi bấm xuống và giữ nút bên trái của mouse, trong khi kéo pointer đi nơi khác).



Ngoài ra bạn cũng có thể sửa đổi các toolbars theo ý thích bằng cách dùng Menu command **View | Toolbars | Customize...**

## Toolbox

Đây là hộp đồ nghề với các công cụ, gọi là controls, mà bạn có thể đặt lên các form trong lúc thiết kế (design). Nếu Toolbox biến mất, bạn có thể display nó trở lại bằng cách dùng menu command **View | Toolbox**. Bạn có thể khiến toolbox display nhiều controls hơn bằng cách chọn **Components...** từ context menu (chọn Toolbox rồi bấm nút phải của mouse để display context menu) hay dùng menu command **Project | Components**. Ngoài việc trình bày Toolbox mặc định, bạn có thể tạo cách trình bày khác bằng cách chọn **Add Tab...** từ context menu và bổ sung các control cho tab từ kết quả.



## Project Explorer

Sẽ liệt kê các forms và các modules trong project hiện hành của bạn. Một **project** là sự tập hợp các files mà bạn sử dụng để tạo một trình ứng dụng. Tức là, trong VB6, khi nói viết một program có nghĩa là triển khai một project.

## Properties window

Liệt kê các đặc tính của các forms hoặc controls được chọn. Một property là một đặc tính của một object chẳng hạn như size, caption, hoặc color. Khi bạn sửa đổi một property bạn sẽ thấy hiệu quả ngay lập tức, thí dụ thay đổi property Font của một Label sẽ thấy Label ấy được display bằng Font chữ

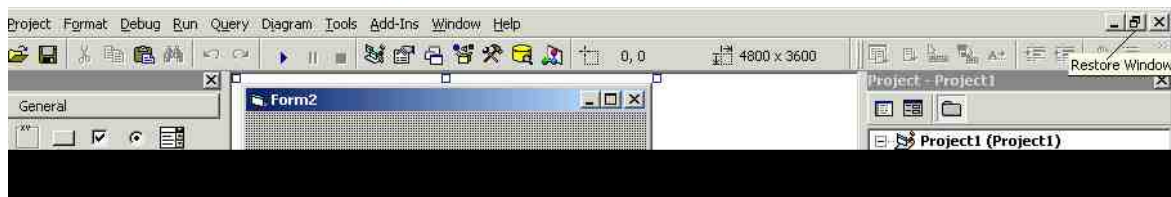
mới. Khi bạn chọn một Property của control hay form trong Properties window, phía bên phải ở chỗ value của property có thể display ba chấm (. . .) hay một tam giác chia xuống. Bấm vào đó để display một dialog cho bạn chọn value. Thí dụ dưới đây là dialog để chọn màu cho property ForeColor của control Label1.

## Form Layout

Bạn dùng form Layout để chỉnh vị trí của các forms khi form hiện ra lần đầu lúc chương trình chạy. Dùng context command **Resolution Guides** để thấy nếu dùng một màn ảnh với độ mịn (resolution) tệ hơn, thí dụ như 640 X 480, thì nó sẽ nhỏ như thế nào.

## Form Designer

Dùng để thiết kế giao diện lập trình. Bạn bổ sung các controls, các đồ họa (graphics), các hình ảnh và một form để tạo sự ma sát mà bạn muốn. Mỗi form trong trình ứng dụng của bạn có designer form riêng của nó. Khi bạn maximise một form designer, nó chiếm cả khu làm việc. Muốn làm cho nó trở lại cỡ bình thường và đồng thời để thấy các form designers khác, click nút Restore Window ở góc bên phải, phía trên.

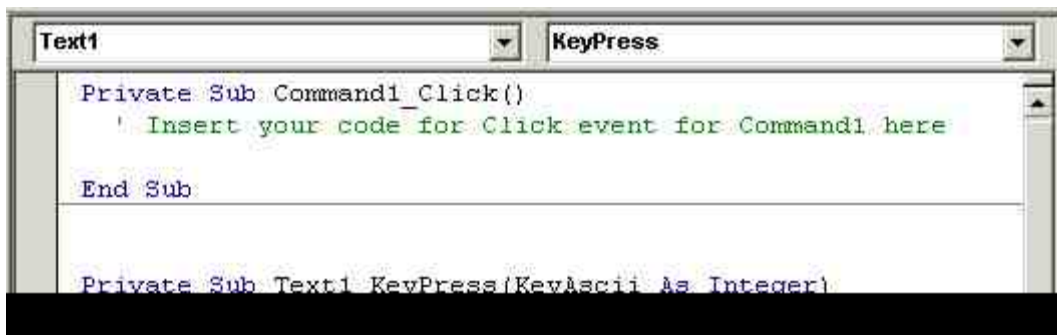


## Immediate Window

Dùng để gỡ rối (debug) trình ứng dụng của bạn. Bạn có thể display dữ kiện trong khi chạy chương trình ứng dụng. Khi chương trình đang tạm ngừng ở một break point, bạn có thể thay đổi giá trị các variables hay chạy một dòng chương trình.

## View Code button

Click lên nút này để xem code của một form mà bạn đã chọn. Window của code giống như dưới đây:



Trong Code window bạn có thể chọn display tất cả Sub của code cùng một lúc như trong hình hay display mỗi lần chỉ một Sub bằng cách click button có hình ba dòng nằm ở góc bên trái phía dưới.

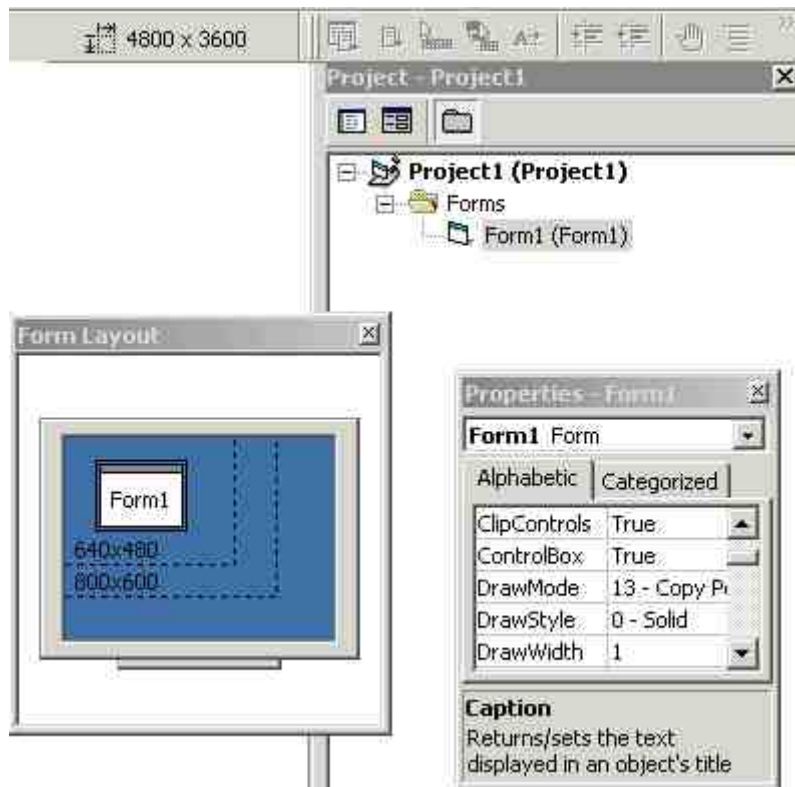


## View form button

Click lên nút này để xem form của một form mà bạn đã chọn.

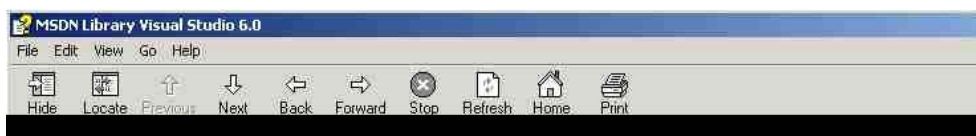
Ghi chú: Nhiều windows trong IDE như Toolbars, Toolbox, Project Explorer .v.v..có thể trôi linh bình (floating) hay đậu ở bên (docked). Bạn có thể thay đổi vị trí chúng bằng cách nắm vào Title Bar của window rồi dời đi. Dĩ nhiên bạn cũng có thể mở rộng hay làm nhỏ một window bằng cách dời một cạnh vertical hay horizontal của nó. Khi để một window lên trên một window khác chúng có thể tìm cách dính nhau.

Trong hình dưới đây, Properties Window và Form Layout đã được kéo ra ngoài cho floating.



## Nhận trợ giúp trong khi đang làm việc

Trong khi lập trình bạn có thể cần tìm hiểu các thông tin liên quan đến các commands, functions .v.v.. của VB6. Bạn có thể khởi động **Microsoft Developer Network | MSDN Library Visual Studio 6.0** từ nút Start, hay click **Help | Contents** từ Menu Bar của VB6, hay chọn một keyword (highlight keyword) rồi ấn F1 để đọc Help.



Nội dung Help bao gồm nhiều đặc điểm được thiết kế để thực hiện việc tìm kiếm thông tin dễ dàng hơn. Bạn có thể dựa trên **Contents** để đọc tài liệu như một quyển sách, **Index** để đọc những đoạn có nhắc đến một keyword hay **Search** để tìm một tài liệu nhanh hơn. Ví dụ, việc gỡ rối thông tin bắt nguồn từ nhiều đặc tính khác nhau phụ thuộc vào loại đề án mà bạn đang làm việc. Các liên kết được mô tả trong phần này thực hiện việc tìm kiếm dễ dàng hơn. Ngoài ra, bạn cũng có thể click **See Also** dưới tiêu đề của chủ điểm để xem các tiêu đề của các chủ điểm mà bạn có thể đi đến hoặc liên hệ đến nhiều thông tin.

## Context Sensitive Help (trợ giúp trong đúng tình huống)

Nhiều phần của VB6 là **context sensitive**, có nghĩa là lúc bối rối chỉ cần ấn nút F1 hoặc highlight keyword rồi nhấn F1 là được thông tin những gì liên hệ trực tiếp với tình huống hiện giờ của bạn. Bạn có thể nhấn F1 từ bất kỳ phần context sensitive nào của giao diện VB6 để display thông tin Help về phần đó. Các phần context sensitive là:

- Các Windows của VB6 như Properties, Code .v.v..
- Các control trong Toolbox.
- Các Object trên một form hoặc Object tài liệu.
- Các đặc tính trong Window Properties.
- Các keywords của VB6
- Các thông báo lỗi (error messages)

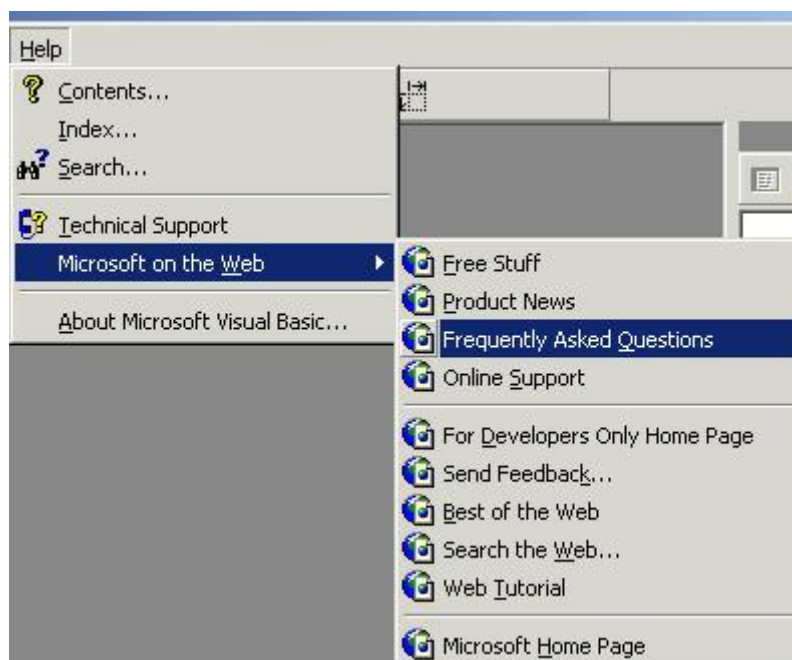
Ngoài ra, trong Help thường có **Example**. Bạn click lên chữ Example để display một thí dụ minh họa cách dùng một function hay property.

## Microsoft on the Web

Web site của Microsoft chứa nhiều thông tin cập nhật cho những người lập trình VB6. Trang chủ Visual Basic đặt tại URL <http://www.microsoft.com/vbasic/>. Thông tin có sẵn tại địa chỉ này bao gồm:

- Cập nhật các đặc tính mới, các phiên bản sản phẩm, các sản phẩm liên hệ, các thuyết trình (seminar) và các hoạt động (event) đặc biệt.
- Thông tin bổ sung trên các đặc tính VB6 chứa trong các bài viết gọi là White Papers, các mách nước (tips) và các trình trợ giáo, nguồn đào tạo.
- Sản phẩm mới tải xuống (download) bao gồm sự cập nhật đến các file chương trình, các cập nhật trợ giúp, các trình điều khiển, và các file liên hệ khác của VB6.

Để truy cập Web site của Microsoft, từ menu Help chọn **Microsoft on the Web** rồi chọn menu con tùy thích như dưới đây.



*Ghi chú: Một số nội dung trên Web site của Microsoft được tối ưu hóa dành cho Microsoft Internet Explorer và không thể display đầy đủ trong một bộ trình duyệt (browser) khác. Do đó bạn nên chỉ dùng Internet Explorer làm browser trên máy bạn mà thôi.*

Tailieu.vn

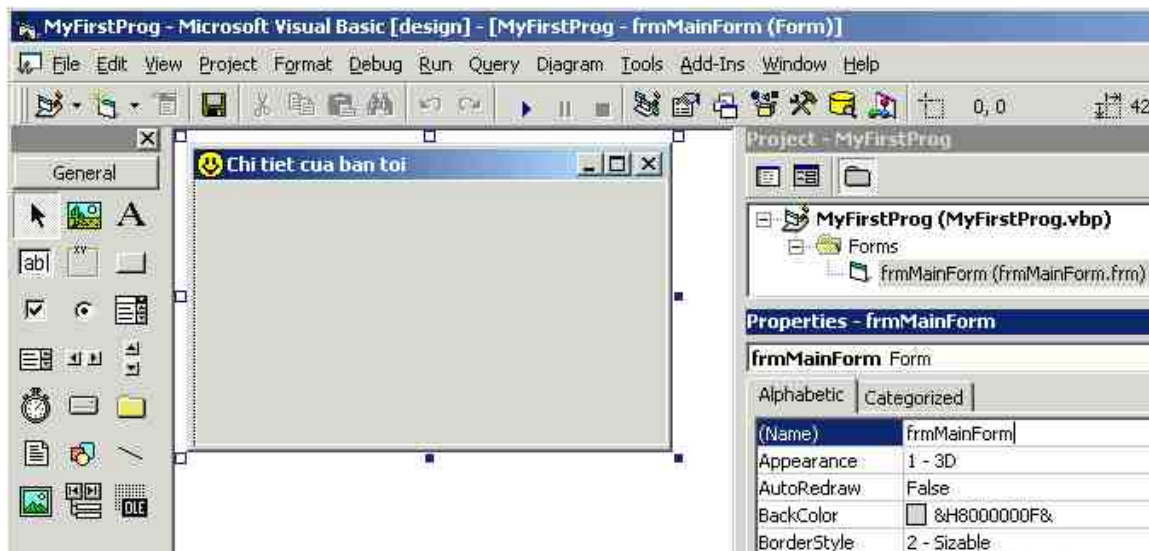


## Chương Hai- Viết chương trình đầu tiên

Bạn đang làm quen với môi trường triển khai lập trình (Integrated Development Environment - IDE) của MS VB6 và rất nóng ruột muốn viết những dòng mã đầu tiên để chào mừng thế giới.

Ta thử ôn lại một số vấn đề mà có lẽ bạn đã biết rồi. Một chương trình Visual Basic gồm có phần mã lập trình và các hình ảnh (visual components). Bạn có thể thiết kế phân hình ảnh bằng cách dùng những đồ nghề (Controls hay Objects) từ Tủ đồ nghề (Toolbox) nằm bên trái. Nếu bạn không thấy cái Tủ đồ nghề thì dùng mệnh lệnh **Menu View|Toolbox** để bắt nó hiện ra.

Khi bạn bắt đầu thiết kế một chương trình bằng cách chọn Standard EXE, môi trường triển khai lập trình (IDE) cho bạn sẵn một Form tên là Form1. Bạn có thể đổi tên (Name) nó trong cái cửa sổ Properties nằm phía dưới bên phải (trong hình dưới đây ta edit Name property của Form1 thành ra frmMainForm). Bạn cũng có thể sửa đề tựa (Title) của form ra cái gì có ý nghĩa hơn bằng cách đổi Caption của form cũng trong cửa sổ Properties (trong hình dưới đây ta edit Caption property của form thành ra "Chi tiet cua ban toi").



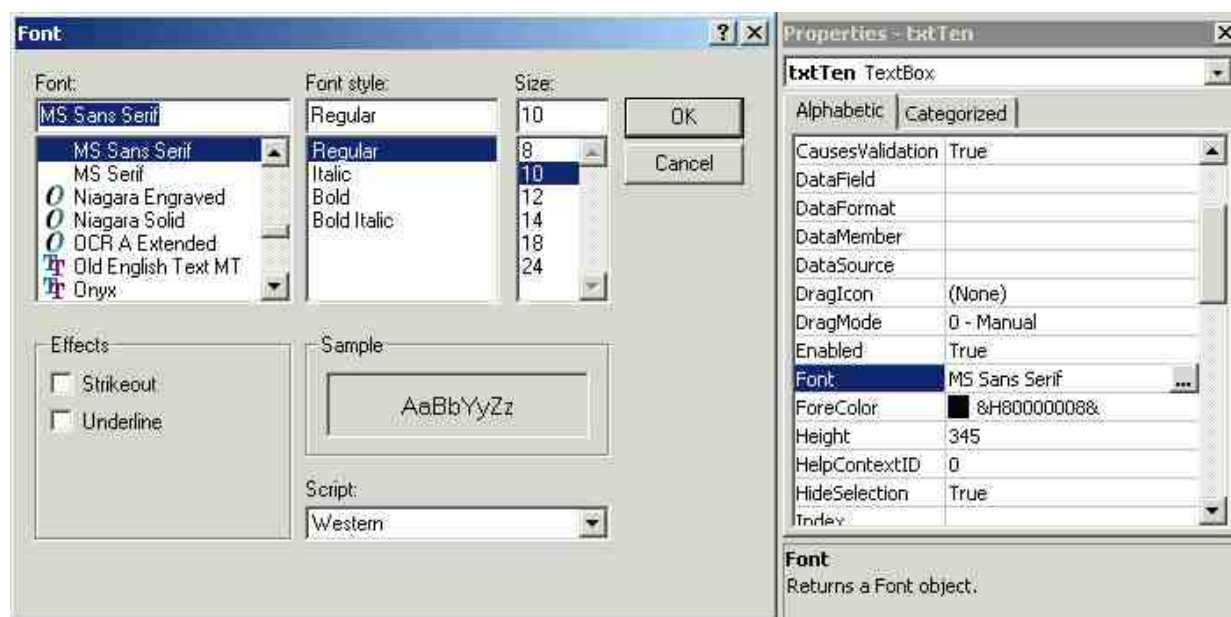
### Sắp đặt các vật dụng lên Form

Muốn đặt một Control lên Form, click hình cái Control trong Toolbox rồi Drag (bấm nút trái của con chuột rồi kéo cho thành hình chữ nhật trước khi buông nút trái ra) con chuột trên Form vẽ thành cỡ của Control. Những Controls bạn sẽ dùng thường nhất từ Toolbox là Label (nhãn), Textbox (hộp để đánh chữ vào) và CommandButton (nút bấm mệnh lệnh).



Trong hình trên ta có ba Label và ba Textbox. Muốn sửa chữ Label1 ra "Ten" thì edit Property Caption. Còn Textbox không dùng Property Caption mà dùng Property Text. Ta cũng có thể thay đổi các Property Caption và Text trong khi chạy chương trình (at run-time). Trong lúc thiết kế (design time) bạn có thể sửa đổi kiểu chữ của những Controls bằng cách edit Property Font của chúng trong

cửa sổ Properties (click bên phải của Property Font trong Properties Window, IDE sẽ pop-up cái Font dialog để bạn lựa chọn những đặc tính của Font như trong hình dưới đây).



Nếu bạn thấy bức mình tại sao cái cỡ chữ tự có (default size) của các Control hơi nhỏ, bạn có thể giải quyết bằng cách sửa cỡ chữ của chính Form cho nó lớn hơn. Vì khi một Control được đặt lên một Form, nó thừa kế cỡ chữ của Form.

Để sắp xếp cho một số Control thẳng hàng với nhau bạn chọn cả nhóm rồi dùng mệnh lệnh Menu **Format|Align|Lefts** .v.v..Nếu bạn chưa biết cách chọn một nhóm Control thì có hai cách. Cách thứ nhất bạn đè nút Shift trong khi click các Control bạn muốn chọn. Cái Control mà bạn chọn sau cùng sẽ là cái chuẩn để các Control khác sẽ làm giống theo. Cách thứ hai là Drag cho sợi dây thun (rubber band) bọc chung quanh các Control. Trong trường hợp các Control này nằm trong một container, thí dụ như một khung (Frame) hay PictureBox, thì bạn phải click Form trước, rồi đè nút Ctrl trong khi Drag rubber band bao các Control.

## Chứa mọi thứ của một dự án VB

Tới đây bạn để ý thấy trong cửa sổ bên phải, phía trên, gọi là Project Explorer, có hình giống như một cái cây (tree) cho thấy ta có một Form trong một Project (dự án). Project là một cách tiện dụng để ta sắp xếp những gì cần thiết cho một dự án. Thường thì một dự án có nhiều Form và có thể cần những thứ khác.

Mỗi Form sẽ được chứa vào đĩa dưới dạng "frmMainForm.frm". Bạn save một form bằng menu command **File | Save formfilename.frm**. Nếu trong Form1 có chứa hình ảnh (thí dụ bạn dùng Properties Window để chọn một icon hình gương mặt cười làm icon cho frmMainForm) thì các hình ảnh của frmMainForm sẽ được tự động chứa trong hồ sơ "frmMainForm.frx". Lưu ý là không nhất thiết tên của hồ sơ (file) mà bạn phải cho biết khi chứa (save) phải giống như tên của Form mà bạn dùng trong chương trình. Tuy nhiên bạn nên dùng cùng một tên cho cả hai để sau này dễ tìm hồ sơ nếu có thất lạc. Theo qui ước thông thường, các Form được đặt tên bắt đầu bằng "**frm**", thí dụ như "frmMainForm".

Khi bạn **save** một Project thì có nghĩa là save tất cả hồ sơ dùng cho dự án, kể cả các Form và một hồ sơ cho chính Project, thí dụ như "MyFirstProg.vbp" ("vbp" là viết tắt chữ Visual Basic Project). Bạn save Vb6 project bằng menu command **File | Save Project**. À, muốn đổi tên Project, bạn click lên hàng trên cùng bên phải trong cửa sổ Project Explorer (Project1 (Project1.vbp)), rồi edit tên của Project trong cửa sổ Properties phía dưới. Bạn nên chứa tất cả những hồ sơ dùng cho cùng một Project trong cùng một tập (Folder/Directory).

Bạn có thể dùng Notepad để mở ra xem chơi, coi trong "frmMainForm.frm" có gì. Bạn sẽ thấy trong ấy gồm có hai phần: phần đầu là diễn tả các Control nằm trong Form, phần còn lại là mã lập trình mà

bạn viết. Bạn cũng sẽ chú ý là các properties mà bạn đã sửa cho các Control đều được ghi lại trong phần đầu nói trên. VB dựa vào phần diễn tả các Control để thiết lập lại (reconstruct) hình ảnh của Form.

```

VERSION 5.00
Begin VB.Form frmMainForm
    Caption       = "Chi tiet cua ban toi"
    ClientHeight  = 2325
    ClientLeft    = 60
    ClientTop     = 345
    ClientWidth   = 4155
    BeginProperty Font
        Name       = "MS Sans Serif"
        Size       = 9.75
        Charset    = 0
        Weight     = 400
        Underline  = 0 'False
        Italic     = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Icon          = "frmMainForm.frx":0000
    KeyPreview    = -1 'True
    LinkTopic     = "Form1"
    ScaleHeight   = 2325
    ScaleWidth    = 4155
    StartUpPosition = 3 'Windows Default
    Begin VB.CommandButton cmdXuat
        Caption       = "&Xuat"
        Height        = 435
        Left          = 150
        TabIndex      = 7
        Top           = 1770
        Width         = 615
    End
    Begin VB.CommandButton cmdViet
        Caption       = "&Viet vao dia"
        Height        = 465
        Left          = 2100
        TabIndex      = 6
        Top           = 1770
        Width         = 1935
    End
    Begin VB.TextBox txtTuoi
        Height        = 375
        Left          = 990
        TabIndex      = 5
        Text          = "29"
    End
End
End
Begin VB.Label lblTen
    Caption       = "&Ten:"
    Height        = 375
    Left          = 210
    TabIndex      = 0
    Top           = 240
    Width         = 495
End
End
Attribute VB_Name = "frmMainForm"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub cmdViet_Click()
    Dim fileNo, myLocalFolder, myFileName
    myLocalFolder = App.Path
    If Right(myLocalFolder, 1) <> "\" Then
        myLocalFolder = myLocalFolder & "\"
    End If
    fileNo = FreeFile
    myFileName = myLocalFolder & "myFriends.txt"
    If Dir(myFileName) <> "" Then
        Open myFileName For Append As #fileNo
    Else
        Open myFileName For Output As #fileNo
    End If
    Print #fileNo, txtTen & ";" & txtDiachi & ";" & txtTuoi
    Close #fileNo
End Sub

Private Sub cmdXuat_Click()
    End
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        SendKeys "{TAB}"
        KeyAscii = 0
    End If
End Sub

```

Sau này, khi đã lão luyện VB, bạn có thể dùng một chương trình tự động chế (generate) ra những hàng diễn tả các Control cho một Form.

Đó là kỹ thuật dùng trong các Wizards của VB để chế một số chương trình khởi đầu cho chúng ta từ các bảng kèm (Template).

## Thêm mã lập trình để xử lý một sự cố

Hầu hết lập trình trong Visual Basic là viết mã để xử lý các sự cố (Event). Thí dụ muốn chấm dứt chương trình, người sử dụng sẽ click nút "Xuat". Để thực hiện điều này trong khi triển khai chương trình bạn doubleClick (click liên tiếp 2 lần) nút "Xuat". VB IDE sẽ viết sẵn cho bạn cái vỏ của một Subroutine:

```

Private Sub cmdXuat_Click()
    End ' Bạn chỉ viết thêm dòng này để kết thúc chương trình
End Sub

```

Đề ý là tên (Name) của nút Xuat là "cmdXuat" ("cmd" là viết tắt chữ CommandButton), VB gắn thêm dấu gạch dưới và Event Click để làm thành tên **cmdXuat\_Click** của Sub, chương trình nhỏ sẽ được xử lý khi người sử dụng click nút Xuat. Chương trình nhỏ hay Subroutine này còn được gọi là Event

Handler cho Event Click. Hàng chữ xanh lá cây là dùng để giải thích cho lập trình viên (gọi là Comment), VB sẽ hoàn toàn không chú ý đến nó khi xử lý Sub cmdXuat\_Click. Comment có nghĩa là chú thích. Trong VB chú thích bắt đầu bằng dấu single quote '. Khi VB thấy dấu này là nó bỏ qua những gì còn lại trên dòng mã.

Là Lập trình viên chuyên nghiệp bạn nên tập thói quen dùng Comment mọi nơi để giúp người khác và chính bạn hiểu chương trình của mình. Nên nhớ là tiền phí tiền để bảo trì một chương trình thì ít nhất là tương đương với số tiền bỏ ra lần đầu để triển khai. Bảo trì có nghĩa là thăm viếng lại chương trình để sửa lỗi (fix bug) và thêm các đặc điểm cho hay hơn (enhancement).

Nói chung hễ bạn làm điều gì bí hiểm hay rắc rối thì làm ơn giải thích rõ ràng.

Nếu muốn cắt một dòng mã VB ra làm hai dòng thì chấm dứt dòng thứ nhất bằng dấu gạch dưới \_.

Tiếp theo, bạn doubleClick nút "Viet vào đĩa" và viết những hàng mã sau:

```
Private Sub cmdViet_Click()  
    Open "myFriends.txt" For Output As #2 ' Mở một hồ sơ để viết ra và gọi là cổng số 2  
    ' Viết vào cổng số 2: Tên, Địa chỉ và Tuổi, ngăn cách nhau bằng dấu chấm phẩy  
    Print #2, txtTen & ";" & txtDiachi & ";" & txtTuoi  
    Close #2 ' Đóng cổng số 2  
End Sub
```

Trong Sub cmdViet\_Click, trước hết ta mở một hồ sơ tên là "**myFriends.txt**" và gọi nó là cổng số 2. Sau khi mở hồ sơ để viết ra ta ráp Tên, Địa chỉ và Tuổi lại, ngăn cách bằng dấu chấm phẩy (;) để đánh dấu nhớ sau này ta muốn gỡ riêng ba thứ ra trở lại. Dấu "&" là operator để ráp (concatenate) hai dòng chữ (text string) lại với nhau.

Print #2 có nghĩa là viết ra cổng số 2, tức là hồ sơ "myFriends.txt".

Thứ chúng ta viết ra cổng 2 là Tên, Địa chỉ và Tuổi (txtTen & ";" & txtDiachi & ";" & txtTuoi).

## Những rắc rối của việc mở một hồ sơ

Cái cổng số 2 ở trên là ta tự chọn (arbitrary). Thật ra muốn gọi cổng số mấy cũng được, miễn là chưa có phần nào khác trong cùng chương trình này đang dùng cổng số ấy. Đây là một cách VB làm việc cho tiện thay vì gọi nguyên một cái tên hồ sơ dài.

Nếu muốn chắc chắn không trùng số cổng với chỗ nào khác, ta có thể làm như sau:

```
fileNo = freefile
```

Rồi thay thế số 2 bằng chữ fileNo trong Sub cmdViet\_Click. **freeFile** là một Function (chương trình nhỏ dùng để tính ra một thứ gì) nhờ VB cấp phát cho một con số đại diện hồ sơ chưa ai dùng.

Chữ **Output** trong câu ( Open "myFriends.txt" For Output As #2 ) dùng ở đây để nói từ CPU (Central Processing Unit) ta muốn "viết ra" một hồ sơ. Khi mở một hồ sơ để viết ra kiểu này thì nếu hồ sơ chưa có nó sẽ được dựng nên (created). Nếu hồ sơ đã có rồi thì nó sẽ bị xóa bỏ (delete) và đồng thời một hồ sơ trống và mới sẽ được dựng nên. Động từ chuyên môn là "viết chồng lên" (**overwrite**).

Nếu ta mở một hồ sơ để "đọc vào" thì dùng chữ "**Input**" thay vì "**Output**". Còn nếu "viết thêm" vào một hồ sơ có sẵn (chớ không overwrite hồ sơ ấy) thì dùng chữ "**Append**" thay vì "Output". Trong trường hợp ấy bạn phải kiểm xem hồ sơ "myFriends.txt" đã có sẵn chưa. Bạn có thể viết như sau:

```
If Dir("myFriends.txt") <> "" then ' Nếu hồ sơ "myFriends.txt" hiện hữu  
    Open "myFriends.txt" For Append As #2 ' Mở một hồ sơ để viết thêm và gọi là cổng số 2  
Else  
    Open "myFriends.txt" For Output As #2 ' Mở một hồ sơ để viết ra và gọi là cổng số 2  
End If
```

Function Dir("myFriends.txt") dùng ở trên sẽ cho ta tên của hồ sơ nếu hồ sơ hiện hữu, ngược lại nó sẽ cho một dòng chữ trống (empty string), biểu hiệu là "". Tại đây, nếu bạn ý bạn sẽ hỏi hồ sơ "myFriends.txt" nằm ở folder nào. Câu trả lời là không biết chắc. Nếu bạn chưa chứa



(save) chương trình vào đĩa (vì mới viết) thì nó nằm ở folder của VB6.EXE. Còn như đã chứa chương trình rồi thì có lẽ nó nằm ở folder của chương trình bạn. Muốn hồ sơ "myFriends.txt" luôn luôn đi cùng với chương trình, bạn có thể làm như sau:

```
MyLocalFolder = App.path ' Lấy folder của chương trình xử lý của bạn
If Right(MyLocalFolder,1) <> "\" then ' Nếu chữ cuối cùng không phải là backslash
    MyLocalFolder = MyLocalFolder & "\" ' thì gắn thêm một backslash ở cuối
End If
' Mở một hồ sơ với tên có folder (full pathname) để viết ra và gọi là cổng số 2
Open MyLocalFolder & "myFriends.txt" For Output As #2
```

Cuối cùng ta đóng hồ sơ lại bằng câu Close #2.. Từ rày VB có thể cấp số 2 để làm cổng cho chỗ khác trong chương trình.

## Default Property của một Control

"txtTen" được dùng ở đây là viết tắt cho "txtTen.text", vì Default Property của một TextBox là text của nó. Default Property của một Control là Property được VB dùng khi bạn chỉ cho tên của Control mà thôi.

Trong khi đó Default Property của Label là Caption.

Vì txtTen được dùng như txtTen.txt để nói đến một dòng chữ, nên trong chương trình ta nhắc đến nó y như một variable (mã số) dùng cho một string. Do đó với qui ước dùng ba chữ đầu "txt" cho tên của một Textbox giúp ta nhận diện ngay nó không phải là một string variable bình thường. Hãy lưu ý sự khác biệt khi gọi một Sub trong hai trường hợp sau:

```
Call CheckmyTextbox (txtDiachi) ' txtDiachi được xem là Textbox trọn vẹn
CheckmyTextbox txtDiachi ' txtDiachi được xem là txtDiachi.text, một dòng chữ
```

## Thứ tự các Control trên một Form

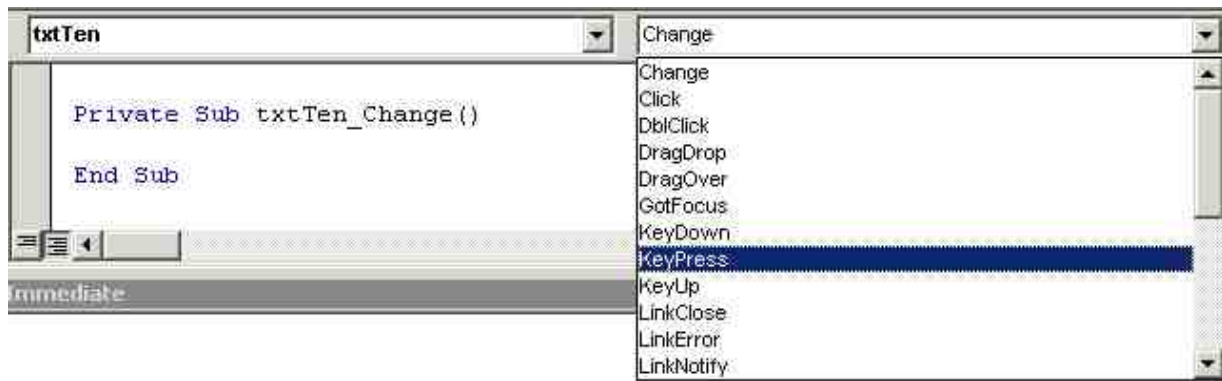
Trong chương trình này ta muốn người xử dụng cho vào dữ kiện theo thứ tự "Tên, Địa chỉ, Tuổi". Khi mới vào, ta muốn cái dấu chớp tắt (cursor) nằm trong txtTen ngay để người xử dụng khỏi mất công click vào Textbox ấy khi muốn mang cursor trở lại đó. Ta nói là txtTen có cái Focus.

Sau khi người xử dụng đã cho tên vào rồi, cô sẽ đánh nút Tab để di chuyển cursor qua Control tiếp theo, mà ta muốn là txtDiachi. Để sắp thứ tự các Control cho sự di chuyển của cursor khi người xử dụng đánh nút Tab ta edit Property TabIndex của các Control. TabIndex bắt đầu bằng số 0. Nhiều khi người xử dụng thích dùng nút Enter thay vì Tab để di chuyển Cursor qua Control tiếp theo, bạn có thể làm như sau cho Textbox txtTen:

```
Private Sub txtTen_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then ' Nếu nút bấm là Enter
        SendKeys "{TAB}" ' giả mạo gọi nút Tab
        KeyAscii = 0 ' Nuốt trọn nút Enter để Windows không còn lo cho nó
    End If
End Sub
```

Cho các Textbox khác như txtDiachi, txtTuoi bạn cũng làm tương tự như vậy. Khi bạn doubleClick txtTen lần đầu để viết mã, VB cho bạn **Private Sub txtTen\_Change()**. Bạn phải click cái Combobox bên phải, phía trên của Code Window, cho nó mở ra và chọn Event KeyPress.





Nếu bạn muốn chương trình mình còn chuyên nghiệp hơn, bạn cho phép người sử dụng bấm nút **Alt+o** (bấm nút **Alt** xuống trong khi bấm nút **o**) để mang Cursor về txtTuoi hay **Alt+d** để mang Cursor về txtDiachi. Muốn thế bạn phải thêm vào dấu "&" ở phía trước các chữ **T**, **D** và **o** trong Caption của các label lblTen, lblDiachi và lblTuoi.

Kể đó bạn phải cho giá trị **TabIndex** của lblTen, txtTen, lblDiachi, txtDiachi, lblTuoi, txtTuoi liên tiếp là 0,1,2,3,4,5. Khi người sử dụng đánh Alt+o, VB sẽ mang Cursor về nhãn lblTuoi, nhưng vì không có chỗ cho nó đáp trong label nên nó phải đáp vào Control kế đó, tức là txtTuoi.

Khi ta đã cho TabIndex của các Control những giá trị kể trên thì khi Form hiện ra Cursor sẽ nằm trong TextBox txtTen vì mặc dầu lblTen có TabIndex nhỏ nhất(0), nó không phải là chỗ Cursor đáp lên được, nên Cursor phải đáp lên textbox có TabIndex value kế đó, tức là 1.

Nếu bạn không muốn Cursor ngừng lại ở một TextBox nào thì edit Property **TabStop** của TextBox đó cho bằng False. Trong trường hợp ấy người sử dụng vẫn có thể click vào TextBox và sửa dòng chữ ở đó được như thường. Nếu bạn thật sự không muốn cho phép người sử dụng sửa gì ở TextBox thì edit Property **Enabled** bằng False hay Property **Locked** bằng True. Khi Enabled của một TextBox bằng False thì TextBox trở nên mờ đi.

Nhân tiện ta edit thêm dấu "&" ở phía trước các chữ X và V trong Caption các CommandButton "Xuat" và "Viet vao dia". Sau này người sử dụng có thể bấm Alt-X coi như tương đương với click nút "Xuat".

Nếu nhờ trong Form bạn có nhiều Textbox quá, đổi nút Enter ra nút Tab cho từng Textbox một thì mất công quá. Bạn có thể làm một cái chung cho cả Form. Tức là nói rằng bạn không cần biết nút Enter vừa mới được đánh ở TextBox nào, bạn cứ nhắm mắt đổi nó ra nút Tab.

Trước hết bạn phải chọn (select) Form rồi edit Property **KeyPreview** của nó thành True. Bạn làm việc này để dẫn Form giựt cái nút người sử dụng đánh (keystroke) trước khi TextBox thấy. Form sẽ trao nút Enter thành Tab rồi lẳng lặng trao cho TextBox. Bạn có thể thay thế tất cả các KeyPress event handler của các TextBox bằng đoạn mã như sau:

```
Private Sub Form_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then ' Nếu nút bấm là Enter
        SendKeys "{TAB}" ' giả mạo gọi nút Tab
        KeyAscii = 0 ' Nuốt trọng nút Enter để Windows không còn lo cho nó
    End If
End Sub
```

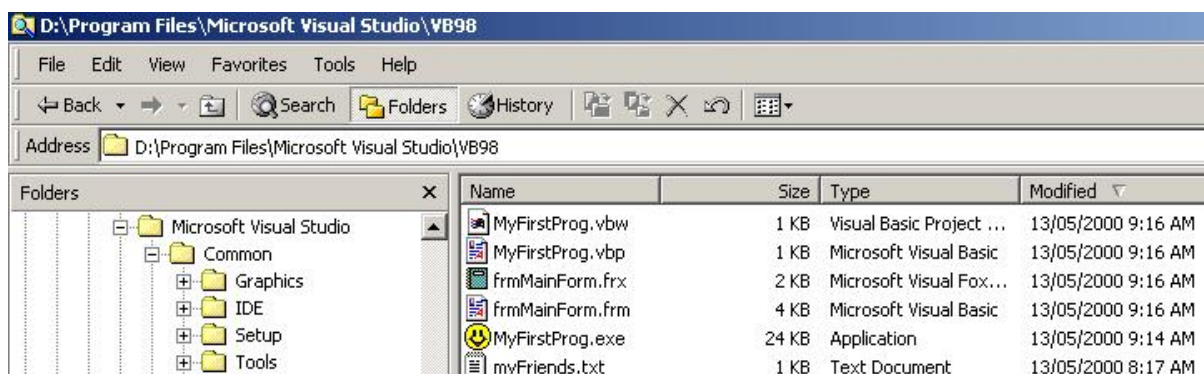
Khi bạn doubleClick lên bất cứ chỗ nào trên Form không có Control nằm, lần đầu để viết mã, VB cho bạn Private Sub Form\_Load(). Bạn phải click cái Combobox bên phải, phía trên của Code Window, cho nó mở ra và chọn Event KeyPress.

## Dem ra trình làng

Để làm thành một hồ sơ áp dụng EXE, bạn dùng mệnh lệnh Menu **File|Make MyFirstProg.exe**. Cho thêm chút hương vị của cuộc đời tôi click Form rồi edit Property Icon, chọn cho nó từ folder:

D:\Program Files\Microsoft Visual Studio\Common\Graphics\Icons\Misc

một icon hình gương mặt cười. Rồi bấm mệnh lệnh Menu **File|Save Project**. Khi dùng Explorer để xem các hồ sơ của MyFirstProg.vbp bạn sẽ thấy như dưới đây:



Đáng lẽ tôi dùng một folder khác thay vì VB98 để chứa dự án MyFirstProg.vbp. Hồ sơ MyFirstProg.vbw là Workspace (chỗ làm việc) dành cho VB, ta không nên động tới. Bạn có thể làm một Shortcut cho MyFirstProg.exe với cái icon hình gương mặt cười đặt lên Desktop để chạy bên ngoài IDE của VB. Có lẽ bạn muốn Download hồ sơ: [MyFirstProg.zip](#), nên chung tất cả các hồ sơ nói trên trong dự án MyFirstProg.vbp. Bây giờ ngay trong VB IDE bạn có thể chạy chương trình bằng cách dùng mệnh lệnh Menu **Run|Start** hay bấm **F5**.



Bạn cũng có thể Click lên dấu tam giác chỉ về bên phải (nút Play của cassette) nằm trong toolbar ngay phía dưới VB menu.

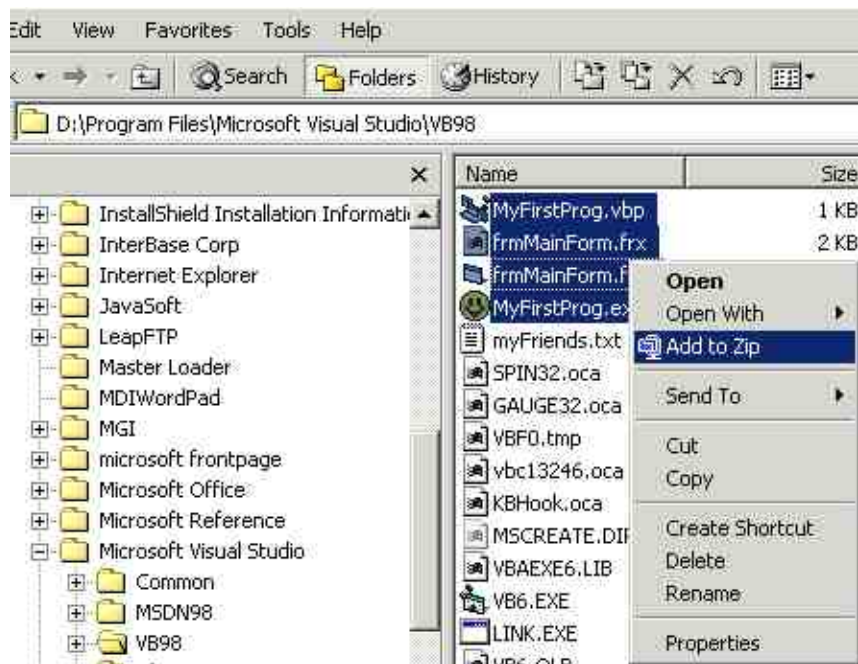


## Cách nén các files trong một folder thành một zip file duy nhất

Để gởi nhiều files bằng cách đính kèm (attach) một Email trên Internet ta cần phải nén các files ấy thành một file duy nhất, gọi là Zip file. Trước hết, trong Window Explorer bạn chọn những files bạn muốn Zip chung lại. Bạn chọn nhiều files bằng cách đè nút **Ctrl** trong khi click lên tên từng file một. Nếu bạn đè lên nút **Shift**, thay vì nút Ctrl, thì cứ mỗi lúc bạn click, Window Explorer sẽ select cả một dãy tên các files nằm giữa tên hai files bạn click mới nhất. Ngoài ra bạn cũng có thể dùng Menu Command **Edit | Select All**, hay **Ctrl+A** để select tất cả các files trong một folder. Đây là trường hợp bạn sẽ dùng khi Zip tất cả các files trong một VB6 project để gởi qua Thầy/Cô.

Sau khi đã select các file rồi, bạn right click lên các file ấy để context menu pop-up. Chọn **Add to Zip**.

Nếu bạn không thấy pop-up command Add to Zip thì là bạn chưa install chương trình Winzip. Trong trường hợp ấy, download Winzip từ Internet và install.



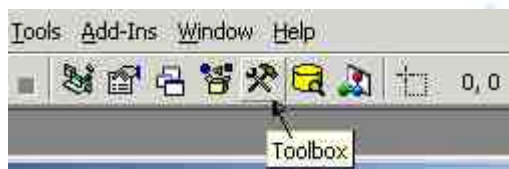
Bây giờ bạn thử khởi động VB6, đi dạo quanh nó để thử biết các phần của VB6 IDE, và thử làm theo như bài này. Nếu có thắc mắc gì thì hỏi Tutor của bạn.

## Chương Ba - Form và các Controls thông thường

Hầu hết các chương trình VB6 đều có ít nhất một Form. Khi ta chạy chương trình, Form này sẽ hiện ra trước hết để ta ra lệnh nó làm chuyện gì. Cái Form trông không chả làm được gì nhiều, nên ta đặt lên Form những controls như Textbox(hộp để đánh chữ vào), Label(nhãn), Commandbutton(nút bấm mệnh lệnh), .v.v.. Các controls cho ta enter các dữ kiện để chương trình dùng xử lý, và các controls cũng hiển thị (display) kết quả cho chúng ta xem.

### Sắp đặt controls lên Form

Ta hãy bắt đầu thiết kế một chương trình mới (New Project) bằng cách chọn Standard EXE, môi trường triển khai lập trình (IDE) cho bạn sẵn một Form tên là Form1. Muốn đặt một Control lên Form, click hình cái Control trong Toolbox rồi Drag (bấm nút trái của con chuột rồi kéo cho thành hình chữ nhật trước khi buông nút trái ra) con chuột trên Form vẽ thành cỡ của Control. Một cách khác để đặt một control lên Form là doubleclick cái Control trong Toolbox, một hình control sẽ hiện ra trên Form. Kế đó bạn dời control đi đến chỗ mình muốn và resize nó. Nếu bất cứ lúc nào bạn không thấy Tủ đồ nghề (Toolbox) nằm bên trái, bạn có thể dùng mệnh lệnh **Menu View|Toolbox** để bắt nó hiện ra. Có một cách khác là click lên toolbox icon trên toolbar chính của VB6.



Nên nhớ rằng Toolbox cũng là một window như các window khác. Khi nó hiện lên rồi bạn có thể nắm (bấm nút trái của con chuột và giữ như vậy chờ không buông ra) title nó để dời đi nơi khác. Bạn có thể đóng nó bằng cách click lên dấu x ở góc phải phía trên. Nếu right click trên Toolbox, nó sẽ display context sensitive menu, trong đó có property dockable (có thể đậu ở bên) . Nếu một window là dockable, sau khi bạn dời nó đi khỏi vị trí docked bình thường của nó, bạn có thể dock nó lại như cũ bằng cách double click lên title của nó.

### Resize và di chuyển control

Khi bạn select một control (click lên nó), chung quanh control sẽ hiện ra resize handle, 8 nút đen dọc theo chu vi của control.



Click lên các nút đen của resize handle, bạn có thể resize control. Có một cách khác để resize control là dùng Shift + ArrowKey. Bấm nút Shift trong khi bấm một arrow key, control sẽ lớn ra hay thu hẹp theo chiều của ArrowKey.

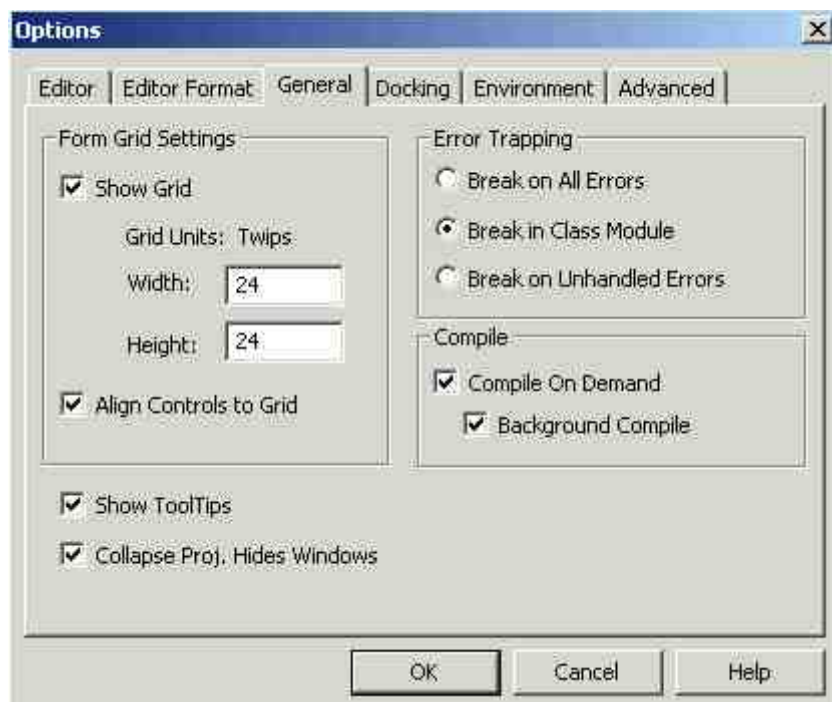
*Lưu ý: Một số control có kích thước tối thiểu, bạn không thể làm cho nó nhỏ hơn được. Thí dụ như Combobox, nó phải cao đủ để display một hàng text.*

Tương tự như thế, bấm nút Ctrl trong khi bấm một arrow key, control sẽ di chuyển theo chiều của ArrowKey.

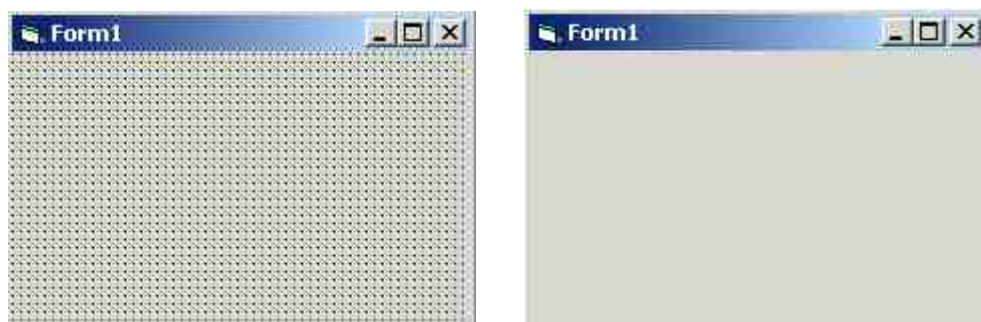
Ngoài ra, nên nhớ rằng trong lúc chương trình chạy (at run-time), trong code ta có thể thay đổi kích thước và vị trí các controls dễ dàng, thậm chí có thể làm cho chúng hiện ra hay biến mất bằng cách sửa đổi value các property left, top, width, height và visible của các controls.

## Alignment Grid

Để giúp bạn sắp đặt ngay ngắn các controls trên một form, VB6 cho bạn Alignment Grid. Nó là những dấu đen của các hàng dọc và xuôi trên form. Bạn có thể làm cho các dấu đen của grid trên form biến mất bằng cách dùng menu command **Tools | Options** để display Option Dialog, kế đó chọn Tag General và clear checkbox "Show Grid":



Bạn cũng có thể nhân dịp này thay đổi khoảng cách chiều rộng (Width) và chiều cao (Height) của các chấm đen của grid. Kích thước nhỏ nhất của Width hay Height là 24. Hãy so sánh hai trường hợp form có và không có Show Grid như dưới đây:



## Control Locking

Một khi bạn đã sắp đặt kích thước và vị trí của các control trên form rồi, rất dễ ta tình cờ thay đổi các đặc tính ấy vì vô ý click lên một control. Do đó VB6 cho ta Menu command **Format | Lock Controls** để khóa chúng lại. Sau khi khóa, cái hình ổng khóa trên menu bị chìm xuống.





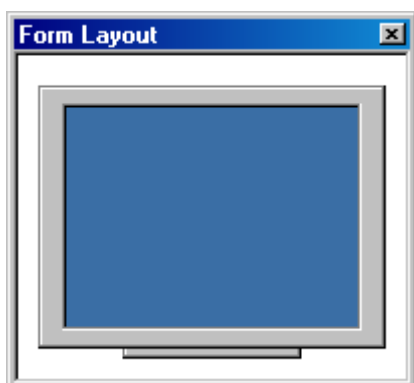
Nếu sau này bạn muốn thay đổi kích thước hoặc vị trí của chúng thì nhớ dùng Menu command **Format | Lock Controls** lại. Sau khi mở khóa, cái hình ổng khóa trên menu hiện ra bình thường.

## Cài đặt các Properties của Form

Nhiều property của một form ảnh hưởng đến diện mạo vật lý (physical appearance) của nó. Property Caption sẽ quyết định text được hiển thị trong title. Nếu Property BorderStyle của form không phải là Sizable thì User không thể resize form at run-time. Property Icon quyết định hình icon được dùng trong title của form, nhất là khi form thu nhỏ (minimized). Nếu bạn không muốn cho phép User minimize hay maximize form thì set value của property MinButton, MaxButton ra False. Nếu property ControlBox là False thì form sẽ không có nút minimize, maximize hay close (x) trên góc phải của nó, đồng thời form cũng không display cả icon bên góc trái title như trong hình dưới đây:



Vị trí đầu tiên (top,left) của form có thể được thay đổi trong design time bằng cách di chuyển hình nhỏ của nó trong window Form Layout:



Property WindowState xác định Form sẽ có kích thước bình thường (normal=0), hay minimized (=1), maximized (=2).

Lưu ý là property Font của Form sẽ được các control nằm trên nó thừa kế. Tức là khi bạn đặt một control lên form, property Font của control ấy sẽ tự động trở nên giống y như của form.

## Vài Event thông dụng của Form

Nhìn từ một phương diện, Form cũng giống như Control. Ta có thể instantiate một form nhiều lần để có nhiều form tương tự nhau. Trong thí dụ dưới đây, ta instantiate Form2 hai lần để có MyForm và YourForm:

```
Private Sub CmdCreateForms_Click()  
    Dim MyForm, YourForm  
    Set MyForm = New Form2  
    MyForm.Caption = "This is My Form"  
    MyForm.Show  
    MyForm.Move 1000, 1000  
    Set YourForm = New Form2  
    YourForm.Caption = "YOUR FORM IS HERE"  
    YourForm.Show  
    YourForm.Move 2000, 2000  
End Sub
```

Một Form cũng có nhiều Events rất hữu dụng.

- **Form\_Initialize:** Event này xảy ra trước nhất và chỉ một lần thôi khi ta instantiate form đầu tiên. Ta dùng Form\_Initialize event để thực hiện những gì cần phải làm chung cho tất cả các instances của form này. F
- **Form\_Load:** Event này xảy ra mỗi lần ta instantiate một form. Nếu ta chỉ dùng một instance duy nhất của một form trong chương trình thì Form\_Load coi như tương đương với Form\_Initialize. Ta dùng Form\_Load event để initialise variables, controls v.v. cho instance này. F  
Bên trong Form\_Load bạn không thể dùng Setfocus cho một control nào trên form vì form chưa hẳn thành hình (ra đời). Muốn làm việc ấy bạn phải delay (trì hoãn) một chút xíu bằng cách dùng Control Timer để đợi cho Form\_Load được hoàn tất. Thí dụ:

```
Private Sub Form_Load()  
    Timer1.Interval = 500  
    Timer1.Enabled = True  
End Sub  
Private Sub Timer1_Timer()  
    Timer1.Enabled = False ' Timer1_Timer only execute once  
    txtName.Setfocus ' Make Tab Cursor start at TextBox txtName  
End Sub
```

- **Form\_Activate:** Mỗi lần một form trở nên active (current) thì nó generate một Activate event. Ta có thể dùng event này để refresh display trên form. F