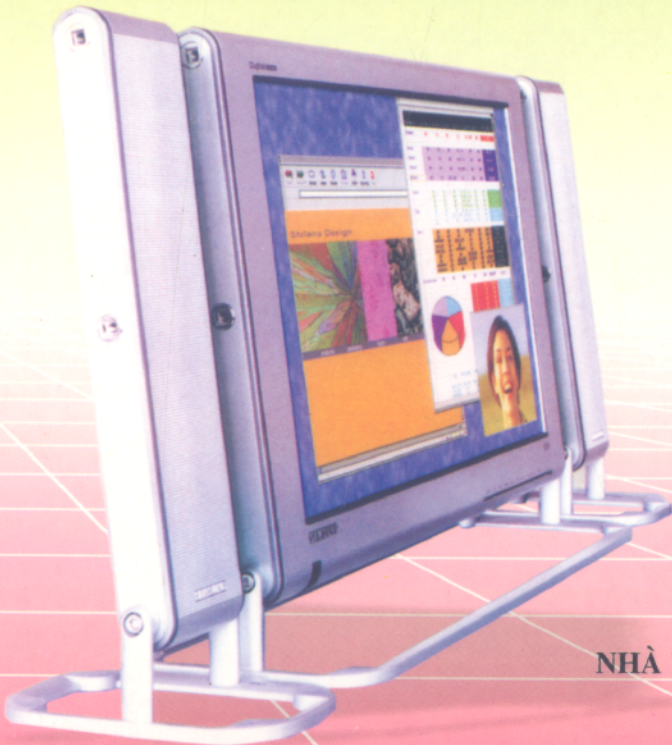


Ban biên dịch
HOÀN VŨ

TỰ HỌC **VISUAL BASIC** **& BÀI TẬP THỰC HÀNH**

- **TÓM TẮT CĂN BẢN CHƯƠNG TRÌNH VISUAL BASIC RẤT DỄ HIỂU**
- **25 BÀI TẬP THỰC HÀNH TỪ DỄ ĐẾN KHÓ**



NHÀ XUẤT BẢN TRẺ

TỰ HỌC
VISUAL BASIC &
BÀI TẬP THỰC HÀNH

Ban biên dịch
HOÀN VŨ

TỰ HỌC
VISUAL BASIC
& BÀI TẬP THỰC HÀNH

NHÀ XUẤT BẢN TRẺ

LỜI NÓI ĐẦU

Trước đây, khi nói đến *Ngôn ngữ lập trình*, chúng ta đều liên tưởng đến các chương trình như Pascal, C++,... Tuy nhiên, với Visual Basic, bạn có thể lập các chương trình giao diện thực tiễn trên nền Windows.

Quyển TỰ HỌC VISUAL BASIC & BÀI TẬP THỰC HÀNH được chia làm hai phần:

- **Phần 1:** tóm tắt các lệnh menu và chức năng của các công cụ nằm trong thanh công cụ General dùng để tạo đối tượng.
- **Phần 2:** gồm những bài tập thực hành từ đơn giản đến phức tạp giúp các bạn làm quen với ngôn ngữ lập trình.

Với những bài tập thực hành mẫu trong quyển sách này, bạn có thể tự thay đổi các thuộc tính cũng như mã lệnh chương trình để thiết kế các đối tượng giao diện theo ý muốn.

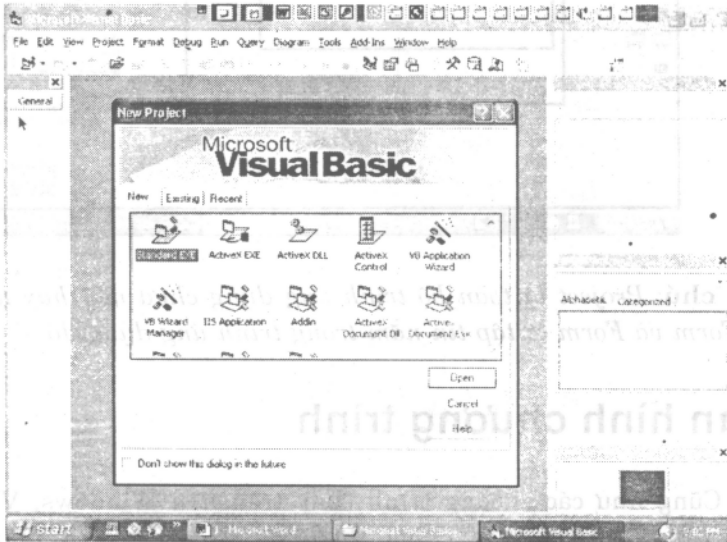
Chúc các bạn thành công và tự thiết kế những đối tượng thú vị và thực tiễn qua phần giao diện từ Ngôn ngữ lập trình Visual Basic.

Trân trọng

Ban biên dịch
HOÀN VŨ

1. Môi trường Visual Basic

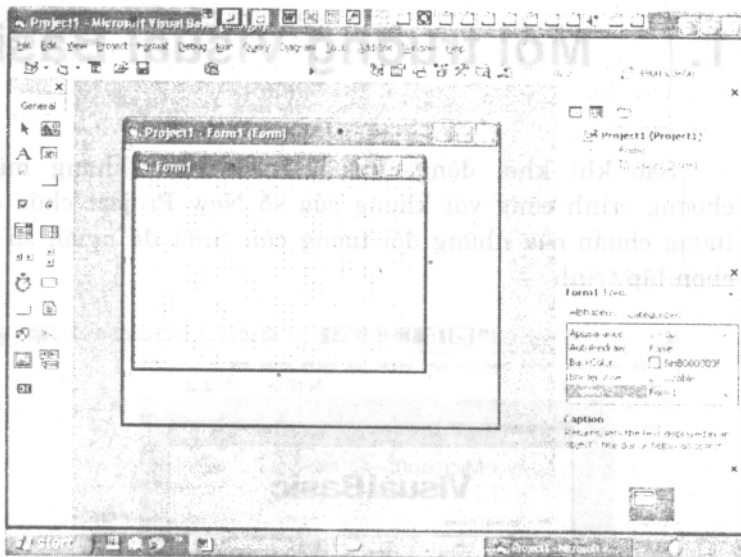
Sau khi khởi động, Visual Basic hiện khung màn hình chương trình cùng với khung cửa sổ New Project chứa các biểu tượng chuẩn của những đối tượng cần thiết để người sử dụng tự chọn lập trình.



Trong khung cửa sổ biểu tượng New Project có ba thước lệnh: New, Existing và Recent để người sử dụng tùy chọn thiết kế mới hoặc xử lý, thay đổi các tập tin chương trình đã tự soạn.

Thông thường, để lập trình một tập tin mới, theo mặc định của chương trình là biểu tượng Standard.EXE được chọn. Đây là một tập tin chuẩn thường được lập trình để điều hoạt chương trình.

Để lập trình một tập tin mới, kích đúp biểu tượng Standard.EXE, màn hình hiện diện một trang thiết kế mới mang tên mặc định **Form1** nằm trong khung trang **Project1**.



Ghi chú: Project là toàn bộ trình ứng dụng chứa một hay nhiều Form và Form là tập tin nằm trong trình ứng dụng đó.

Màn hình chương trình

Cũng như các chương trình chạy trên nền Windows, Visual Basic cũng không thoát khỏi thông lệ. Ngoài thanh tiêu đề, menu, công cụ được bố trí theo hàng ngang nằm theo ba dòng đầu, Visual Basic còn có một thanh công cụ General nằm theo chiều dọc màn hình – giống như các chương trình đồ họa. Đây là những công cụ được dùng để tạo các nút lệnh, các biểu tượng cho lập trình.

Chiếm gần trọn giữa khung màn hình là nơi thiết kế các đối tượng với tên mặc định là **Form1**, nằm trong khung cửa sổ thiết kế **Project**. Bên phải khung cửa sổ thiết kế đối tượng là ba khung cửa sổ chi tiết: Khung cửa sổ trên cùng là khung cửa sổ Project Manager, liệt kê tên của đề án cùng với mẫu biểu thiết kế. Giữa là khung cửa sổ Properties liệt kê thuộc tính gán cho

mẫu biểu thiết kế và khung cửa sổ cuối cùng Layout, trình bày nội dung của mẫu biểu thiết kế.

Thanh công cụ General

Chứa các hộp công cụ hỗ trợ cho việc thiết kế các đối tượng trong khung cửa sổ Form nằm trong khung cửa sổ Project.

Công cụ	Tên	Chức năng
	Pointer	Chọn đối tượng trong Form
	PictureBox	Chèn hình ảnh trong Form
	Label	Gán nhãn đối tượng
	TextBox	Tạo khung ký tự
	Frame	Tạo khung đối tượng
	CommandButton	Tạo nút lệnh
	CheckBox	Tạo nút chọn đối tượng
	OptionButton	Tạo nút tùy chọn
	ComboBox	Tạo khung đối tượng xổ
	ListBox	Tạo khung danh mục xổ



HScrollBar

Tạo thanh cuộn ngang



VScrollBar

Tạo thanh cuộn dọc



Timer

Tạo thời gian



DriveListBox

Tạo khung danh mục ổ đĩa



DirListBox

Tạo khung danh mục thư mục



FileListBox

Tạo khung danh sách tập tin



Shape

Vẽ đối tượng



Line

Kẻ đường thẳng



Image

Du nhập hình ảnh bitmap



Data

Tạo khung dữ liệu



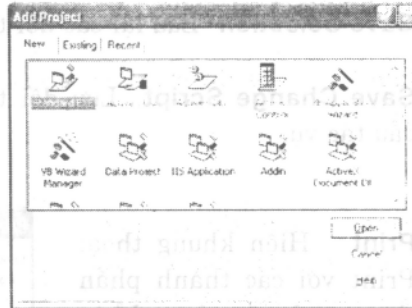
OLE

Tạo khung nổi và nhúng đối tượng

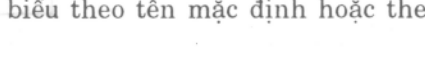
Tóm lược các lệnh menu

File

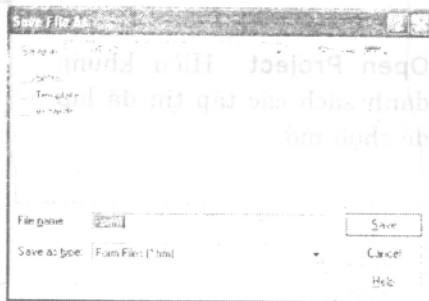
New Project Hiện khung danh mục New Project chứa các biểu tượng đề án tự chọn để tạo mới.



Save Project Lưu tập tin để an theo tên cũ.



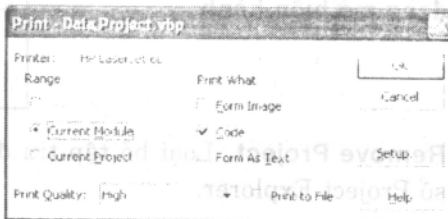
Save Form1 As Hiện khung thoại lưu tập tin mẫu biểu theo tên mặc định hoặc theo tên mới.



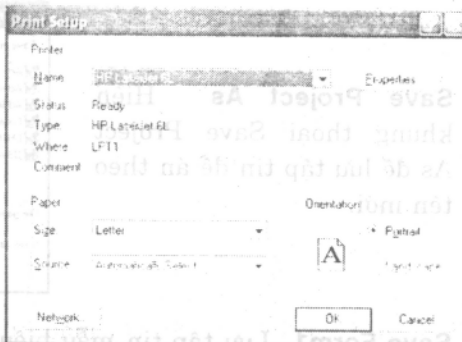
Save Selection Lưu lại các đối tượng đã chọn.

Save Change Script Lưu đối tượng mẫu đã thay đổi theo nhu cầu tác vụ.

Print Hiện khung thoại Print với các thành phần thuộc tính tùy chọn để in tập tin hiện hành.



Print Setup Hiện khung thoại Print Setup để cài đặt máy in theo thuộc tính tùy chọn cho phù hợp.



Make Project1.exe Hiện khung thoại Make Project chứa các bước lệnh để đặt tên mới cùng với các thuộc tính cho tập tin điều hoạt chương trình để án.



Make Project Group Hiện khung thoại để thực hiện việc kết nhóm các tập tin để án để tạo thành một tập tin để án duy nhất.

Exit Thoát khỏi chương trình Visual Basic.

Edit

Undo Insert Text Hủy bỏ tác vụ chèn nội dung mã lệnh đã được thực hiện trước đó.

Can't Redo Không thể trả lại quy trình thực hiện lệnh sau khi đã khôi phục tất cả lệnh tác vụ trước đó.

Cut Cắt đối tượng đã chọn đưa vào Clipboard.

Copy Sao chép đối tượng đã chọn đưa vào Clipboard.

Paste Dán đối tượng từ Clipboard vào vị trí con trỏ hiện hành.

Paste Link Dán đối tượng kết nối đã được chép từ Clipboard vào vị trí con trỏ hiện hành.

Remove Loại bỏ đối tượng đã chọn.

Delete Xóa đối tượng đã chọn. Lệnh có chức năng tương tự như phím Del (Delete) từ bàn phím.

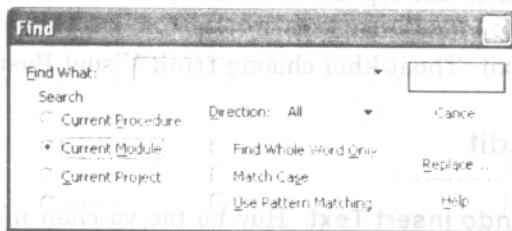
Delete Table from Database Xóa bảng số liệu từ hệ cơ sở dữ liệu.

Select All Chọn tất cả nội dung mã lệnh trong khung cửa sổ Code hoặc tất cả đối tượng trong khung cửa sổ Form.

Select All Columns Chọn tất cả các cột trong bảng dữ liệu.

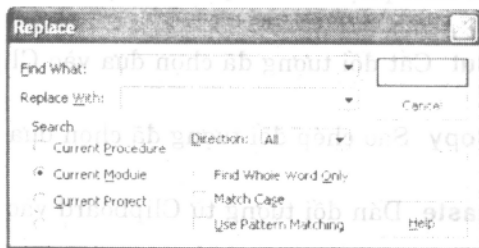
Table Kẻ bảng dữ liệu

Find Hiện khung thoại Find để truy tìm một từ, câu lệnh trong khung ký tự Find What cùng thuộc tính đã chọn.



Find Next Truy tìm một từ, câu lệnh, kế tiếp đã khai báo trong khung thoại Find.

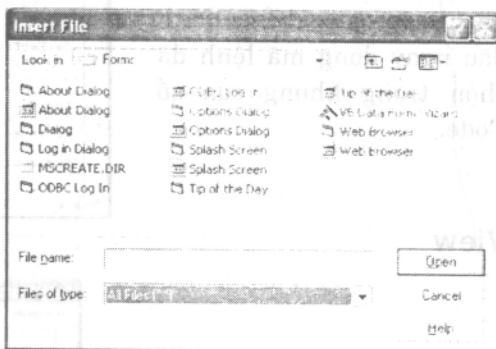
Replace Thay thế một từ, câu lệnh đã truy tìm trong Find What bằng một từ, câu lệnh đã gán trong Replace With cùng thuộc tính đã chọn.



Indent Đẩy chuỗi ký tự sang phải theo khoảng cách mặc định của chương trình trong khung cửa sổ Code.

Outdent Kéo chuỗi ký tự sang trái theo khoảng cách mặc định của chương trình trong khung cửa sổ Code

Insert File Hiện khung danh sách Insert File để chọn tập tin chèn vào vị trí con trỏ trong khung cửa sổ Code hiện hành.



List Properties/Methods Hiện khung danh mục liệt kê các thuộc tính hoặc phương pháp để chèn vào vị trí con trỏ trong khung cửa sổ Code.

List Contants Hiện khung danh mục liệt kê các hằng để chèn vào vị trí con trỏ trong khung cửa sổ Code.

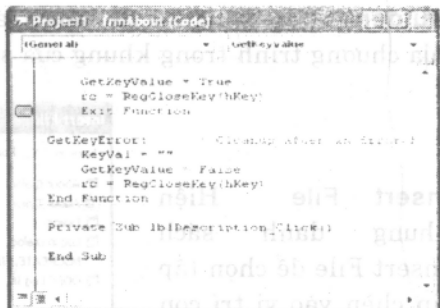
Quick Info Hiện khung thông tin quan sát nội dung của đối tượng được lập trình.

Parameter Info Hiện khung thông tin về các tham số đã khai báo trong khung cửa sổ Code.

Complete Word Lệnh chỉ tác dụng khi chưa kết thúc câu lệnh trong khung cửa sổ Code và con trỏ sẽ nhấp nháy ngay phần còn thiếu để lập trình viên hoàn tất câu lệnh.

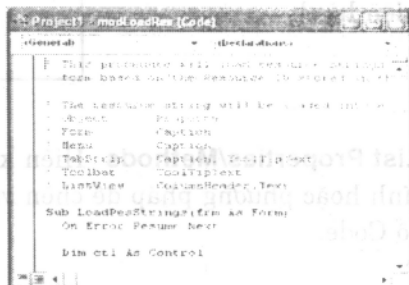
Go To Row Chuyển con trỏ đến dòng mã lệnh cần xử lý trong khung cửa sổ Code theo lệnh chọn từ menu xổ.

Bookmarks Đánh dấu hoặc xóa các điểm đánh dấu ngay dòng mã lệnh đã chọn trong khung cửa sổ Code.



View

Code Hiện khung cửa sổ Code với những nội dung mã lệnh đã lập trình.

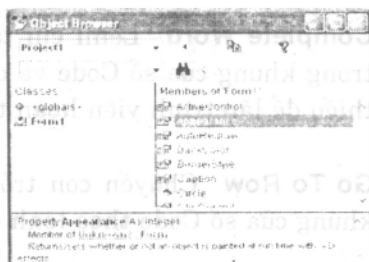


Object Hiện chi tiết đối tượng đã chọn hiện hành.

Definition Xác lập thông tin ngay vị trí con trỏ trong khung cửa sổ Code.

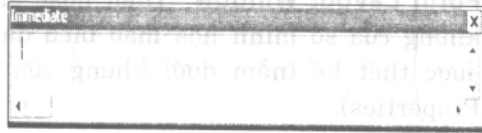
Last Position Chuyển con trỏ đến vị trí cuối cùng đã được thực hiện trong khung cửa sổ Code.

Object Browser Hiện khung danh mục Object Browser với những khung cửa sổ liệt kê danh mục cùng thuộc tính các đối tượng gán vào khung cửa sổ mã lệnh.

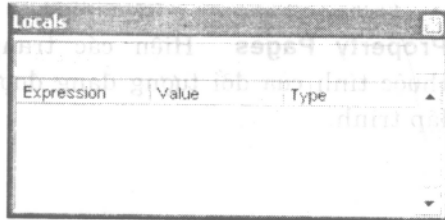


Immediate Window

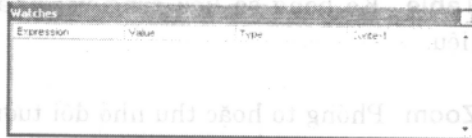
Hiện khung cửa sổ về thông tin nội dung của đối tượng đã lập trình trong khung cửa sổ Code.



Locals Window Hiện khung cửa sổ Locals liệt kê các câu lệnh cùng giá trị và thể loại của đối tượng đang được lập trình.

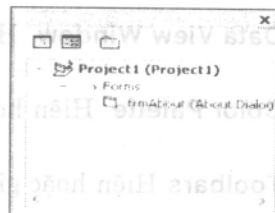


Watch Window Hiện khung cửa sổ Watch trình bày các thông tin về biểu thức, thể loại, giá trị và nội dung của tập tin đang được lập trình.

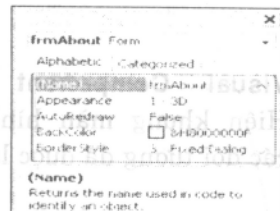


Call Stack Lệnh gọi trình tự các đối tượng trong khung cửa sổ lập trình.

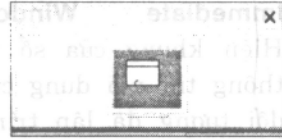
Project Explorer Hiện hoặc giấu khung cửa sổ Project Manager.



Properties Window Hiện hoặc giấu khung cửa sổ thuộc tính (nằm dưới khung cửa sổ Project Manager).



Form Layout Window Hiện hoặc giấu khung cửa sổ minh họa mẫu biểu đang được thiết kế (nằm dưới khung cửa sổ Properties).



Property Pages Hiện các trang thuộc tính của đối tượng đang được lập trình.



Table Kê bảng số liệu cùng với nhãn thuộc tính của các cột số liệu.

Zoom Phóng to hoặc thu nhỏ đối tượng theo tỷ lệ từ menu xổ.

Show Panes Hiện hoặc giấu các khung cửa sổ đối tượng.

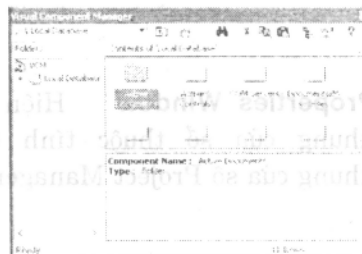
Toolbox Hiện hoặc giấu hộp công cụ.

Data View Window Hiện khung cửa sổ số liệu.

Color Palette Hiện hoặc giấu bảng thuộc tính màu.

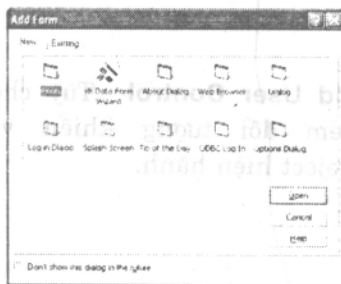
Toolbars Hiện hoặc giấu thanh công cụ trên màn hình.

Visual Component Manager
Hiện khung màn hình quản lý các đối tượng đã được lập trình.



Project

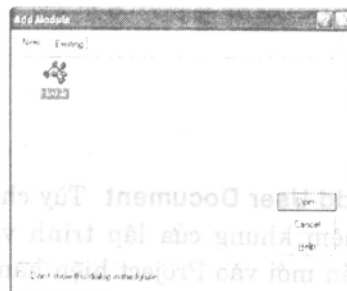
Add Form Hiện khung danh mục tùy chọn thêm một mẫu biểu mới vào mẫu biểu hiện hành.



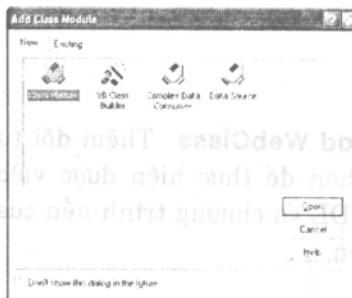
Add MDI Form Hiện khung danh mục tùy chọn thêm mẫu biểu MDI vào mẫu biểu hiện hành.



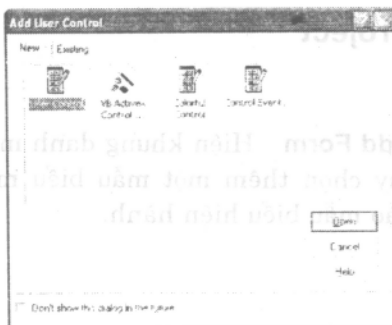
Add Module Thêm khối mã chương trình vào Project hiện hành.



Add Class Module Tùy chọn thêm cấp khối đối tượng vào Project hiện hành.



Add User Control Tùy chọn thêm đối tượng khiển vào Project hiện hành.



Add Property Page Thêm trang thuộc tính mới hoặc đang có sẵn vào tập tin Project hiện hành.

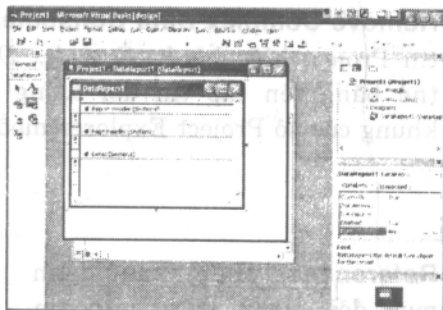


Add User Document Tùy chọn thêm khung cửa lập trình văn bản mới vào Project hiện hành.

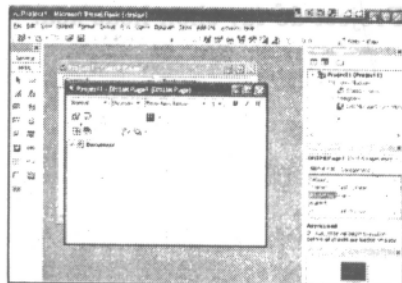


Add WebClass Thêm đối tượng Web vào Project hiện hành. Từ nhên để thực hiện được việc này, bạn phải dùng tập tin để án DDL và chương trình nền của máy chủ từ phiên bản Serve 3.0 trở lên.

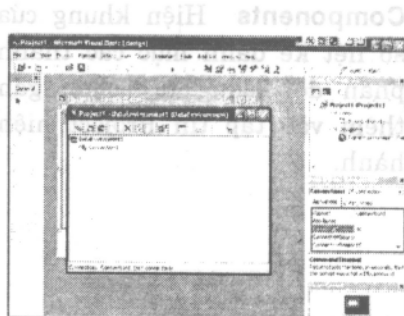
Add Data Report Thêm mẫu biểu báo cáo dữ liệu vào Project hiện hành.



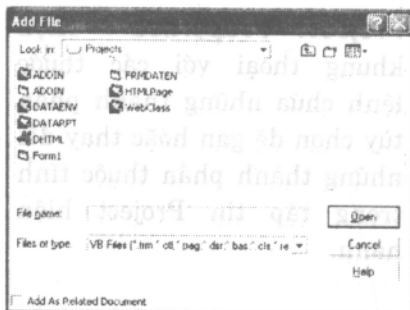
Add DHTML Page Thêm trang DHTML vào Project hiện hành.



Add Data Environment Thêm trang môi trường dữ liệu vào Project hiện hành.

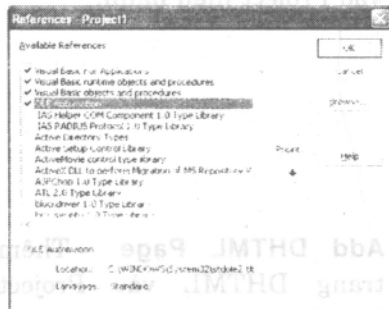


Add File Tùy chọn tên tập tin để thêm vào tập tin Project hiện hành.



Remove Complex Data Consumer.cls Loại bỏ tập tin trong tập tin Project hiện hành. *Complex Data Consumer.cls* sẽ được thay thế bằng tên tập tin đối tượng lập trình đã kích chọn trong khung của sổ Project Explorer muốn loại bỏ.

References Hiện khung danh mục đối tượng trong tập tin Project để tùy chọn hoặc xóa.



Components Hiện khung của sổ liệt kê danh mục các thành phần đối tượng tùy chọn gắn thêm vào tập tin Project hiện hành.



Project1 Properties Hiện khung thoại với các thuộc tính chứa những thành phần tùy chọn để gắn hoặc thay đổi những thành phần thuộc tính trong tập tin Project hiện hành.



Format

Align Chính nội dung trong cột theo lệnh chọn vị trí từ menu xổ.

Make Same Size Chỉnh cột số liệu có cùng cỡ.

Size to Grid Chỉnh cột số liệu theo khung lưới.

Horizontal Spacing Định khoảng cách đối tượng theo chiều ngang với giá trị mặc định.

Vertical Spacing Định khoảng cách đối tượng theo chiều dọc với giá trị mặc định.

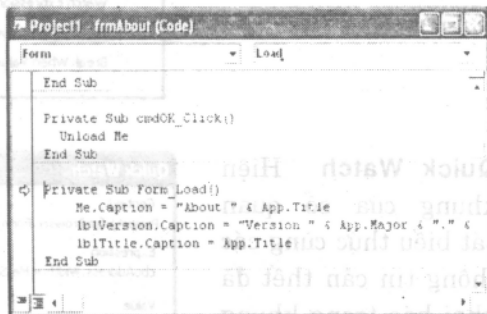
Center in Form Di chuyển đối tượng vào giữa khung mẫu biểu theo lệnh chọn từ menu xổ.

Order Chuyển đối tượng về trước hoặc sau một đối tượng khác.

Lock Controls Khóa đối tượng để tránh vô tình bị di chuyển hoặc thay đổi hoặc mở khóa để xử lý.

Debug

Step Into Hiện vị trí lỗi trong khung cửa sổ Code.

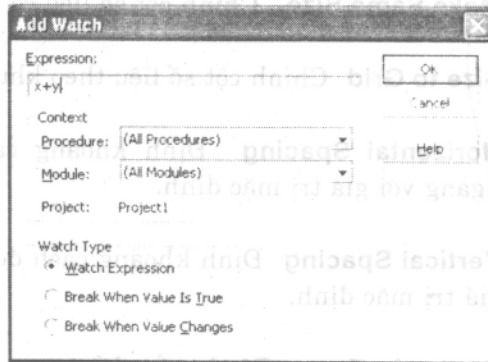


Step Over Hiện tất cả vị trí lỗi trong khung cửa sổ Code.

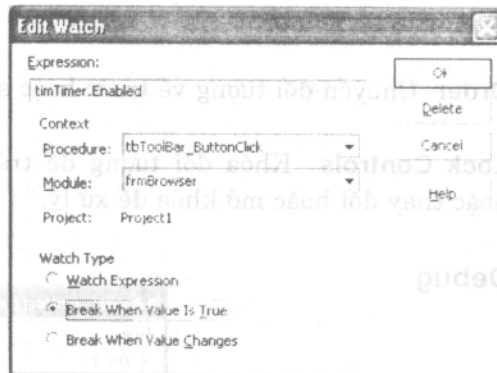
Step Out Trở về đối tượng thiết kế.

Run To Cursor Chuyển con trỏ đến vị trí lỗi trong khung cửa sổ Code.

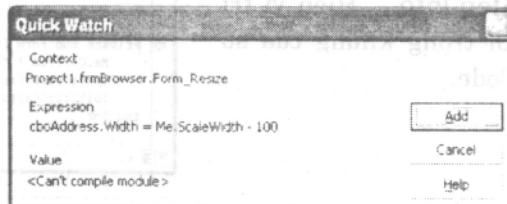
Add Watch Hiện khung thoại Add Watch để thêm biểu thức cùng những thuộc tính khác trong khung cửa sổ Code.



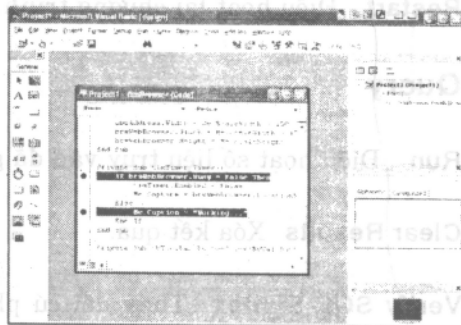
Edit Watch Hiện khung thoại Edit Watch để thay đổi biểu thức đã khai báo trong khung cửa sổ Code.



Quick Watch Hiện khung cửa sổ quan sát biểu thức cùng các thông tin cần thiết đã khai báo trong khung cửa sổ Code.



Toggle Breakpoint Đánh dấu riêng ngay vị trí con trỏ trong khung cửa sổ Code.



Clear All Breakpoints Xóa tất cả các điểm đã đánh dấu trong khung cửa sổ Code.

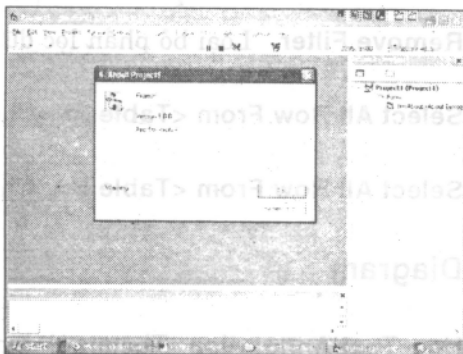
Set Next Statement Gán câu lệnh kế tiếp.

Show Next Statement Hiện câu lệnh kế tiếp.

Run

Start Điều hoạt chương trình sau khi lập trình.

Start With Full Compile
Điều hoạt chương trình sau khi lập trình cùng với khung cửa sổ biên dịch.



Break Ngắt quy trình điều hoạt chương trình.

End Kết thúc quy trình điều hoạt chương trình.

Restart Điều hoạt lại chương trình.

Query

Run Điều hoạt số liệu truy vấn đã gán trong bảng số liệu.

Clear Results Xóa kết quả.

Verify SQL Syntax Thay đổi cú pháp SQL trong khung cửa sổ Code.

Group By Kết nhóm các câu lệnh.

Change Type Thay đổi thể loại của đối tượng.

Add To Output Thêm đối tượng vào câu lệnh xuất.

Sort Ascending Sắp xếp dữ kiện theo thứ tự tăng dần.

Sort Decending Sắp xếp dữ kiện theo thứ tự giảm dần.

Remove Filter Loại bỏ phần lọc dữ kiện.

Select All Row From <Table A> Chọn tất dòng từ <Bảng A>.

Select All Row From <Table B> Chọn tất dòng từ <Bảng B>.

Diagram

New Text Annotation Thêm nội dung thông tin mới.

Set Text Font Chọn phong ký tự.

Add Related Tables Thêm bản số liệu quan hệ.

Show Relationship Labels Hiện các nhãn có liên quan giữa các đối tượng.

Modify Custom View Thay đổi khung cửa sổ.

View Page Breaks Quan sát đường ngắt trang.

Recalculate Page Breaks' Tính lại giá trị ngắt trang.

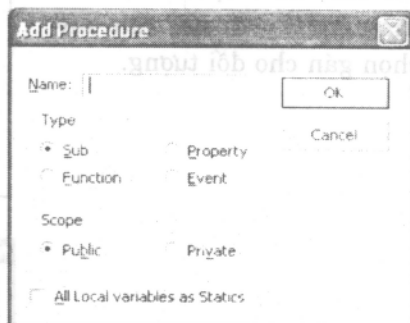
Arrange Selection Bố trí các thành phần đối tượng đã chọn.

Arrange Tables Sắp xếp các bảng số liệu.

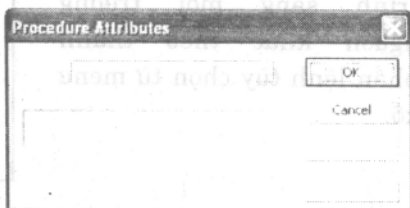
Autosize Selected Tables Tự động chỉnh kích cỡ bảng số liệu đã chọn.

Tools

Add Procedure Thêm thủ tục mới vào vị trí đã chọn trong khung cửa sổ Code.



Procedure Attributes Gán thuộc tính cho thủ tục.



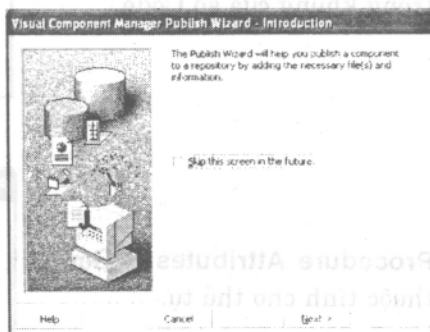
Menu Editor Thay đổi thành phần thuộc tính chương trình menu.



Options Hiện khung thoại với các thuộc lệnh chứa các thành phần thuộc tính tùy chọn gán cho đối tượng.



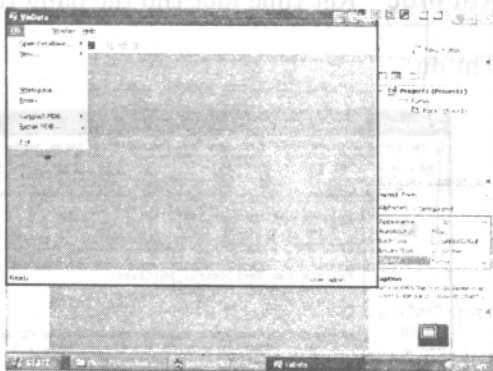
Publish Xuất đối tượng lập trình sang môi trường nguồn khác theo thành phần lệnh tùy chọn từ menu xổ.



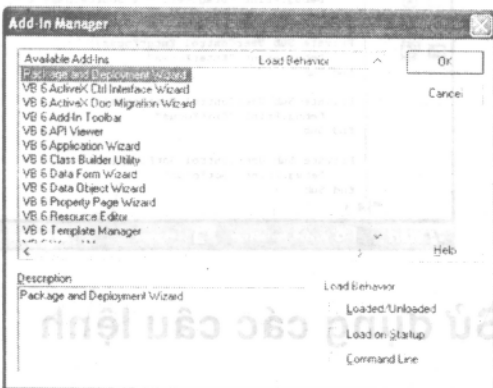
Add-Ins

Visual Data Manager

Hiện khung màn hình quản lý các nguồn dữ liệu từ những chương trình dữ liệu khác muốn nhập vào khung màn hình Code.



Add-In Manager Hiện khung danh mục các đối tượng muốn thêm vào chương trình.



Lập trình Visual Basic

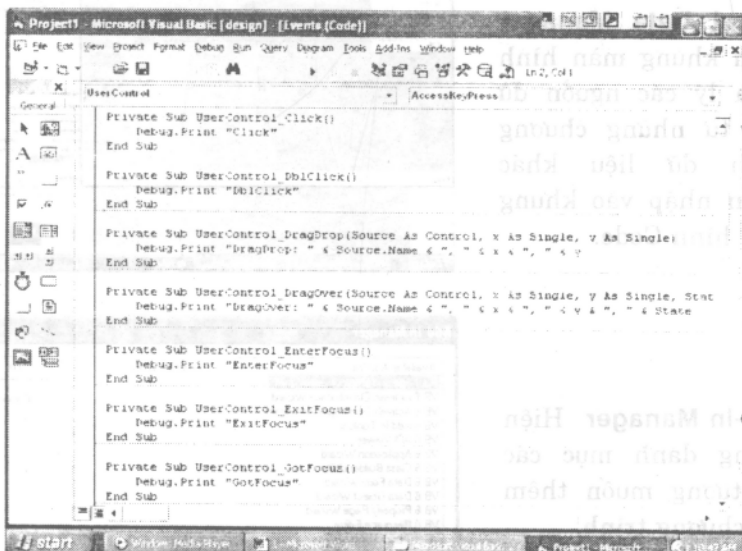
Việc lập trình được chia làm ba phần:

Thủ tục (Procedure) Thủ tục hoặc phương pháp của sự kiện hoặc hành động và thường được gán theo tên của đối tượng được lấy từ thanh công cụ đưa vào khung cửa sổ Form và đặt tên trong khung cửa sổ Properties.

Biểu thức (Expression) Gồm các câu lệnh, biểu thức để thực thi sự kiện của thủ tục hoặc phương pháp và kết thúc bằng từ khóa End của câu lệnh đó.

Kết thúc Kết thúc một thủ tục bằng từ khóa End.

Thí dụ,



Sử dụng các câu lệnh

Các câu lệnh được lập trong chương trình Visual Basic thường được kết thúc bằng từ khóa END.

Tuy nhiên khi cần thực hiện một quy trình kế tiếp một quy trình trước đó, chuyển tới lui một hành động, hoặc thực hiện một hành động mang tính đệ quy, chúng ta thường thực hiện các câu lệnh và trong Visual Basic có rất nhiều câu lệnh và tùy từng trường hợp, chúng ta gán những câu lệnh cho phù hợp.

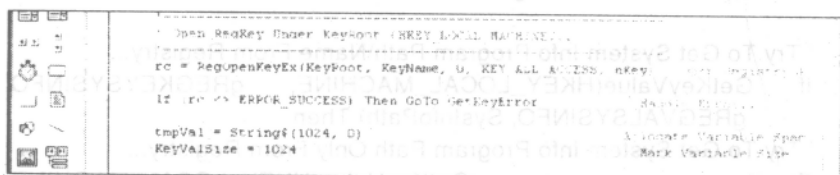
Câu lệnh điều kiện IF ... THEN

Câu lệnh điều kiện được thực hiện nếu điều kiện lệnh được thỏa mãn thì đưa ra kết quả.

Câu lệnh được thực hiện theo cú pháp:

If (câu lệnh 1) Then (thực hiện chức năng)
End if

Thí dụ:

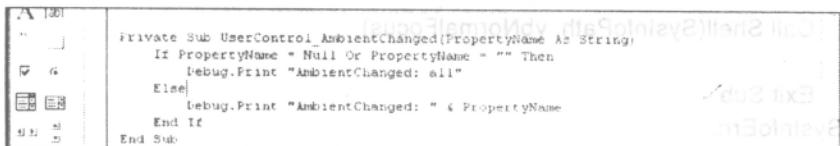


IF ... THEN ... ELSE

Câu lệnh này cũng tương tự như câu lệnh If...Then, nhưng nếu câu lệnh 1 không thỏa mãn hoặc không xảy ra thì sẽ thực hiện theo câu lệnh kế tiếp. Câu lệnh này được thực hiện theo cú pháp:

If <câu lệnh 1> Then <kết quả> Else <câu lệnh 2>
End If

Thí dụ,



Ghép lệnh

Ghép lệnh (nest) là đặt một câu lệnh điều kiện vào một câu lệnh điều kiện khác. Trong Visual Basic, việc ghép lệnh các câu lệnh **IF** được chấp nhận nếu một câu lệnh IF được kết thúc bằng một khóa lệnh **End IF**.

Thi dụ,

```
Public Sub StartSysInfo()  
    On Error GoTo SysInfoErr  
  
    Dim rc As Long  
    Dim SysInfoPath As String  
  
    ' Try To Get System Info Program Path\Name From Registry...  
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,  
        gREGVALSYSINFO, SysInfoPath) Then  
        ' Try To Get System Info Program Path Only From Registry...  
    Elseif GetKeyValue(HKEY_LOCAL_MACHINE,  
gREGKEYSYSINFOLOC, gREGVALSYSINFOLOC, SysInfoPath) Then  
        ' Validate Existence Of Known 32 Bit File Version  
        If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then  
            SysInfoPath = SysInfoPath & "\MSINFO32.EXE"  
  
            ' Error - File Can Not Be Found...  
        Else  
            GoTo SysInfoErr  
        End If  
        ' Error - Registry Entry Can Not Be Found...  
    Else  
        GoTo SysInfoErr  
    End If  
  
    Call Shell(SysInfoPath, vbNormalFocus)  
  
    Exit Sub  
SysInfoErr:  
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly  
End Sub
```

Câu lệnh lặp

Câu lệnh lặp được dùng để quy trình thực hiện nhiều lần.
Trong Visual Basic, lệnh lặp gồm có hai câu lệnh:

DO WHILE

Câu lệnh thực hiện một hành động trong khi một hành động kia có giá trị logic là TRUE.

Thí dụ,

```
Private Sub Timer1_Timer()  
a = 1  
Do While a < 24  
Number = Number + Str(a)  
a = a + 1  
Loop  
End Sub
```

DO UNTIL

Câu lệnh có chức năng kiểm tra liên tục một hành động và chức năng đó sẽ ngưng cho đến khi bắt gặp một hành động mang giá trị TRUE.

Thí dụ,

```
Private Sub Timer1_Timer()  
a = 24  
Do Until a + 2  
Debug.Print "try again!"  
Loop  
End Sub
```

FOR NEXT

Câu lệnh có chức năng thực hiện theo tính liên tục một hành động dựa theo số lượng ký tự hoặc chữ đề trong một tập hợp.

Thí dụ,

```
For i = Len(tmpVal) To 1 ' Convert Each Bit  
KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1)))  
' Build Value Char. By Char.
```

```
Next
KeyVal = Format$("&h" + KeyVal)
```

FOR STEP NEXT

Câu lệnh có chức năng thay đổi số bước trong câu lệnh chu trình For Next.

```
For i = Len(tmpVal) To 1 Step -1      ' Convert Each Bit
    KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1)))
' Build Value Char. By Char.
Next
KeyVal = Format$("&h" + KeyVal)
```

Câu lệnh trường hợp CASE

Câu lệnh có chức năng ấn định một trường hợp xảy ra dựa theo giá trị từ một thông số của một tập hợp đã được ấn định.

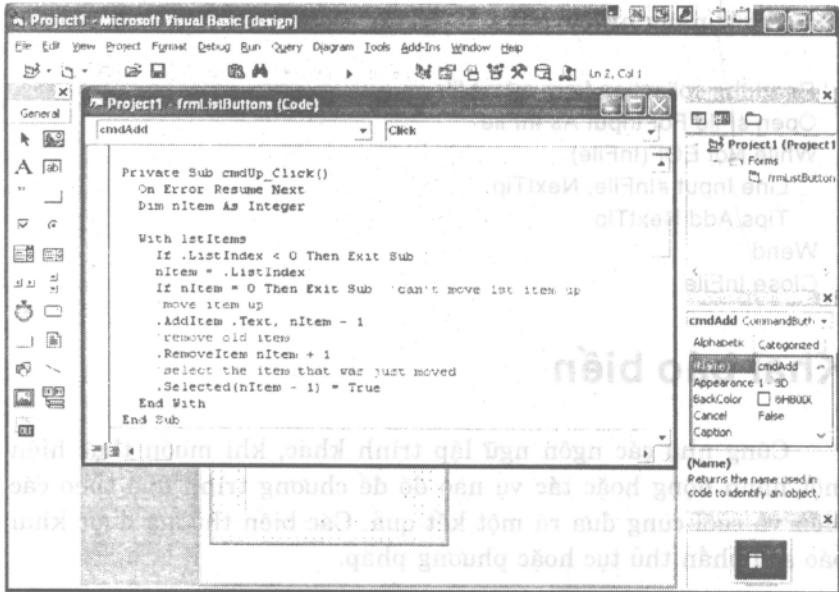
Thí dụ,

```
Select Case KeyValType      ' Search Data Types...
    Case REG_SZ              ' String Registry Key Data Type
        KeyVal = tmpVal      ' Copy String Value
    Case REG_DWORD           ' Double Word Registry Key Data Type
        For i = Len(tmpVal) To 1 Step -1      ' Convert Each Bit
            KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1))) ' Build Value
            Char. By Char.
        Next
        KeyVal = Format$("&h" + KeyVal)      ' Convert Double Word To
                                                String
End Select
```

Câu lệnh kết hợp WITH

Có chức năng thực hiện một hành động kết hợp với một hành động khác cho đến khi điều kiện hành động đó được thỏa mãn.

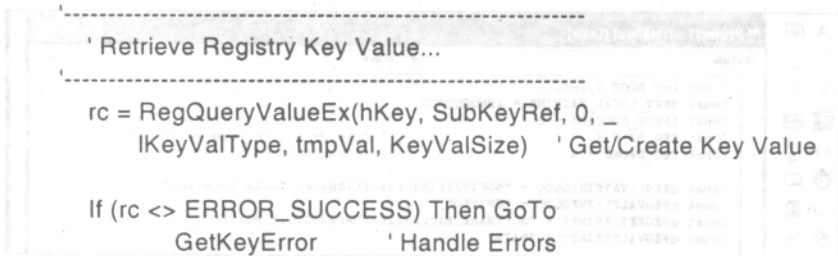
Thí dụ,



Câu lệnh chuyển GO TO

Có chức năng thực hiện một hành động chuyển đến một vị trí nào đó theo khai báo hằng trong câu lệnh.

Thí dụ,



tmpVal = Left\$(tmpVal, InStr(tmpVal, Chr(0)) - 1)

Câu lệnh Wend

Câu lệnh Wend được dùng để thay câu lệnh lặp While-End và có chức năng như vòng lặp. Thí dụ,

```
' Read the collection from a text file.  
Open sFile For Input As InFile  
While Not EOF(InFile)  
    Line Input #InFile, NextTip  
    Tips.Add NextTip  
Wend  
Close InFile
```

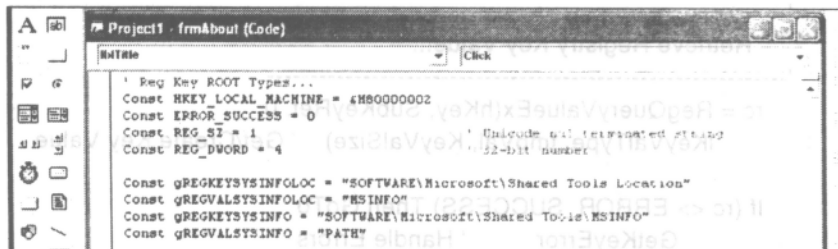
Khai báo biến

Cũng như các ngôn ngữ lập trình khác, khi muốn thực hiện một hành động hoặc tác vụ nào đó để chương trình dựa theo các biến và cuối cùng đưa ra một kết quả. Các biến thường được khai báo sau phần thủ tục hoặc phương pháp.

Khai báo hằng (CONST)

Hằng (Constant) là một giá trị không đổi và giá trị hằng có thể là một chuỗi giá trị, một con số và đứng sau dấu gán =. Việc khai báo hằng được thực hiện bằng từ khóa CONST.

Thí dụ,



Khai báo loại dữ liệu DATA TYPE

Khai báo loại dữ liệu là khai báo dạng nhất định của một biến. Các khai báo và xác lập loại dữ liệu này được thực hiện bằng khóa hàm DIM (Dimension). Trong Visual Basic, khai báo loại dữ liệu bao gồm các giá trị:

Integer

Khai báo Integer là một khai báo trị nguyên. Thí dụ, trong chương trình thực hiện chức năng kích nút phải mouse, phần mã lệnh được khai báo như sau:

```
Private Sub cmdRightAll_Click()  
    On Error Resume Next  
    Dim i As Integer  
    For i = 0 To lstAll.ListCount - 1  
        lstSelected.AddItem lstAll.List(i)  
    Next  
    lstAll.Clear  
    lstSelected.ListIndex = 0  
End Sub
```

Long

Khai báo biến chuỗi dài có giá trị từ 2 byte trở lên, có thể đó là một chuỗi số hoặc chuỗi ký tự. Thí dụ,

```
Public Function GetKeyValue(KeyRoot As Long, KeyName As  
String, SubKeyRef As String, ByRef KeyVal As String) As Boolean  
    Dim i As Long           ' Loop Counter  
    Dim rc As Long          ' Return Code  
    Dim hKey As Long        ' Handle To An Open Registry Key  
    Dim hDepth As Long
```

String

Giá trị khai báo thuộc dạng chuỗi. Chuỗi có thể thuộc dạng số hoặc ký tự như tên, số ID,... Thí dụ, trong phần khai báo hàm

kích nút lệnh nhấn UNG DUNG để hiện chuỗi ký tự và chuỗi số đăng ký.

```
Private Sub cmdUNG_DUNG_Click()  
Public Function GetKeyValue(KeyName As String, SubKeyRef As String,  
ByRef KeyVal As String) As Boolean ' Data Type Of A Registry Key  
    Dim tmpVal As String ' Tempory Storage For A Registry Key Value  
                                ' Size Of Registry Key Variable  
    -----  
    ' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}  
    -----  
    rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS,  
        hKey) ' Open Registry Key  
    If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError ' Handle Error...  
  
    tmpVal = String$(1024, 0) ' Allocate Variable Space  
    KeyValSize = 1024  
End Function
```

Boolean

Khai báo Boolean để thực hiện hai chức năng: hoặc là TRUE, hoặc FALSE. Thí dụ,

```
Option Explicit  
  
Public LoginSucceeded As Boolean  
  
Private Sub cmdCancel_Click()  
    'set the global var to false  
    'to denote a failed login  
    LoginSucceeded = False  
    Me.Hide  
End Sub
```

Array

Mảng (Array) là chuỗi ba chiều của các biến và có thể hình dung chúng là những dòng và cột. Trước khi áp dụng khai báo mảng, trước hết chúng được khai báo sau từ khóa DIM. Thí dụ,

Dim SODIEM(1 To 4) As String

SODIEM(1) = 5

SODIEM(2) = 10

SODIEM(3) = 15

SODIEM(4) = 20

Khai báo thủ tục

Trong việc khai báo biến cho chương trình, bạn có thể khai báo cho thủ tục hoặc phương pháp bằng một trong hai dạng: Private (riêng) hoặc Public (chung).

Private

Khai báo Private có tính giới hạn trong một phạm vi đã định trước theo biểu thức trong phần thân chương trình. Nếu vượt quá giới hạn, chúng sẽ gây nên lỗi. Thí dụ, trong việc khai báo tải mẫu biểu vào giữa khung màn hình được giới hạn theo biểu thức khai báo chương trình:

```
Private Sub Form_Load()  
    'center the form  
    Me.Move (Screen.Width - Me.Width) / 2,  
            (Screen.Height - Me.Height) / 2  
End Sub
```

Public

Ngược lại với khai báo Private là khai báo Public. Với khai báo này, giá trị của biến được áp dụng cho cả chương trình. Thí dụ, trong chương trình tô màu cho đối tượng, đối tượng bất kỳ được chọn sẽ được tô bằng màu theo lệnh khai báo, trừ phi chọn màu khác.

```
Public Property Get Color() As OLE_COLOR  
    Color = Shape1.FillColor  
End Property
```

Từ khóa đối chiếu ME

Đây là một từ khóa lập sẵn áp dụng vào một tên hoặc một đối tượng bất kỳ khi thực hiện một hành động. Thí dụ, trong câu lệnh khai báo chỉnh lại kích thước của đối tượng, từ khóa ME được gán với một giá trị chiều rộng tối đa và nếu vượt quá giá trị đó, những giá trị còn lại không tác dụng.

```
Private Sub Form_Resize()  
    If Me.Width < 3000 Then Me.Width = 3000  
    SizeControls imgSplitter.Left  
End Sub
```

Toán tử

Các toán tử sử dụng trong Visual Basic gồm hai loại:

Toán pháp:

Là những ký hiệu toán học bao gồm:

Ký hiệu	Chức năng	Ký hiệu	Chức năng
=	Bằng	>	Lớn hơn
<	Nhỏ hơn	>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng	<>	Không bằng

Logic

Được áp dụng để đưa ra kết quả khi một trong hai điều kiện được thỏa mãn.

Toán tử	Nghĩa	Chức năng
Or	Or	Đưa ra giá trị TRUE khi thỏa mãn một điều kiện bất kỳ.
And	and	Đưa ra giá trị TRUE khi tất cả điều kiện được thỏa mãn.

Xor	Exclusive Or	Đưa ra giá trị TRUE khi chỉ có một điều kiện được thỏa mãn.
Not	not	Đảo nghĩa của một điều kiện.
Eqv	Equivalent	Đưa ra giá trị TRUE khi tất cả điều kiện là True hoặc False.
Imp	Impication	Tất cả điều kiện phải là True hoặc False, hoặc có thể bao gồm một trị nguyên.

Đặt tên đối tượng

Khi thiết kế một chương trình, điều quan trọng là cách đặt thuộc tính tên cho đối tượng khi kéo các công cụ đối tượng từ thanh công cụ. Visual Basic áp dụng quy tắc chặt chẽ trong việc đặt tên và đây là một quy luật bắt buộc. Điều này có nghĩa là khi nhìn vào ký tự đầu sẽ biết được kiểu nút lệnh áp dụng, thí dụ, cmdExit, txtKytu, mnuHELP,...

Dưới đây là những khóa từ được đặt trước các tên nút lệnh:

Cmd	CommandButton
Frm	Form
Lbl	Label
Txt	Textbox
Opt	OptionButton
Cls	Class
Lst	Listbox
Mnu	Menu

Sử dụng ký tự &

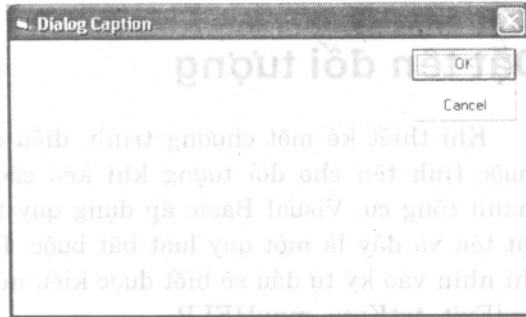
Ký tự & thường được đặt trước ký tự trong khung thuộc tính Caption để tạo phím nhanh cho đối tượng đó và khi hiện trong nút lệnh Command hoặc nhãn Label, ký tự & sẽ được thay bằng dấu gạch dưới cho ký tự nằm sau nó. Thí dụ, muốn tạo phím nhanh cho lệnh EXIT, nhập **&EXIT** và lệnh sẽ được đổi thành **EXIT** trong nút lệnh.

2. Bài tập thực hành

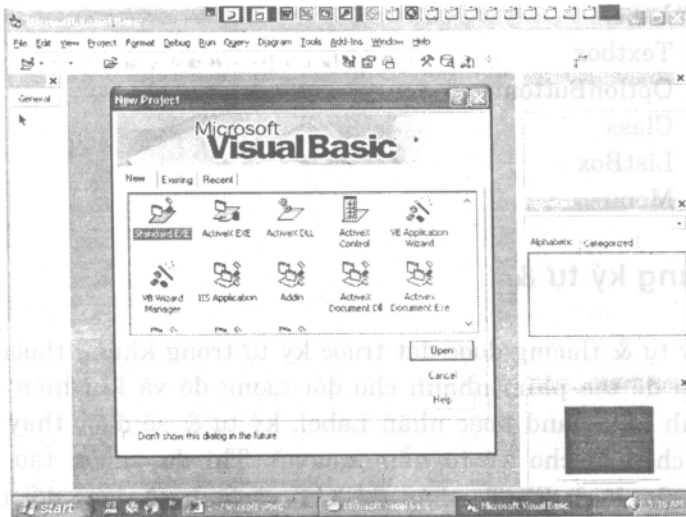
Tạo các mẫu biểu (Form)

Bài 1: Tạo khung thoại Dialog Caption

Trong phần bài tập này, bạn thiết kế một khung thoại mang tên Dialog Caption gồm hai nút lệnh OK và Cancel như hình minh họa:



1. Khởi động chương trình Visual Basic. Màn hình hiện khung của sổ chương trình cùng với các biểu tượng tạo đối tượng trong khung cửa sổ New Project.



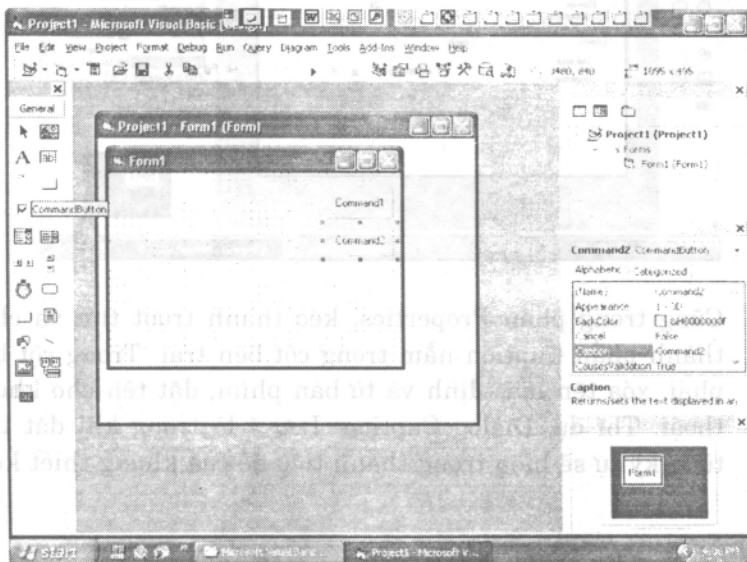
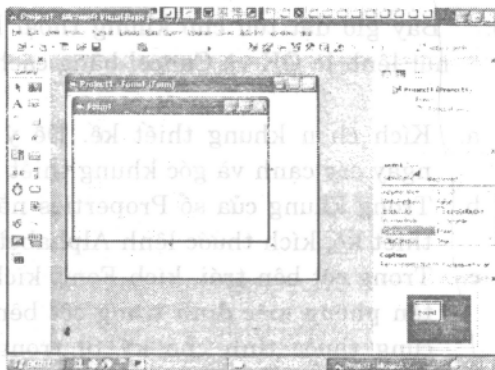
2. Kích đúp biểu tượng Standard. EXE để mở khung mẫu biểu thiết kế.

3. Từ thanh công cụ General, kích chọn công cụ CommandButton.

4. Chuyển con trỏ mouse vào khung thiết kế Form1,

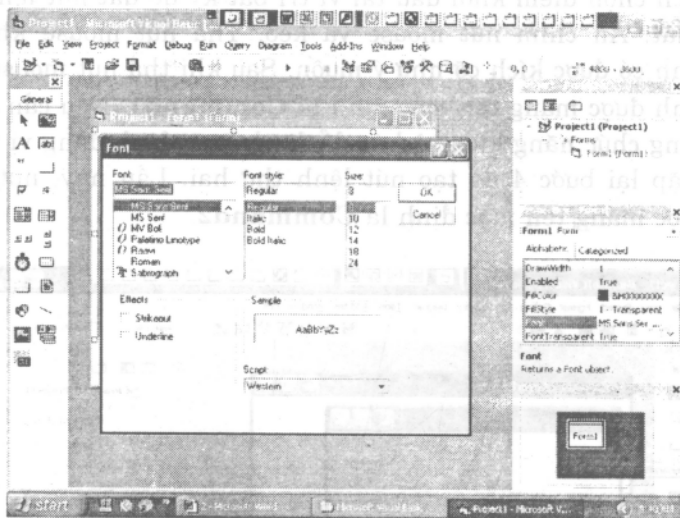
kích chọn điểm khởi đầu tại vị trí bất kỳ để đặt nút lệnh thứ nhất. Ấn chìm nút mouse và kéo. Thả nút mouse khi nút lệnh có được kích cỡ như ý muốn. Sau khi thả nút mouse, nút lệnh được mang tên mặc định là **Command1**. Nếu cần thiết, dùng chức năng kích và kéo để đặt lại nút lệnh đúng vị trí.

5. Lặp lại bước 4 để tạo nút lệnh thứ hai. Lần này, nút lệnh được mang tên mặc định là **Command2**.

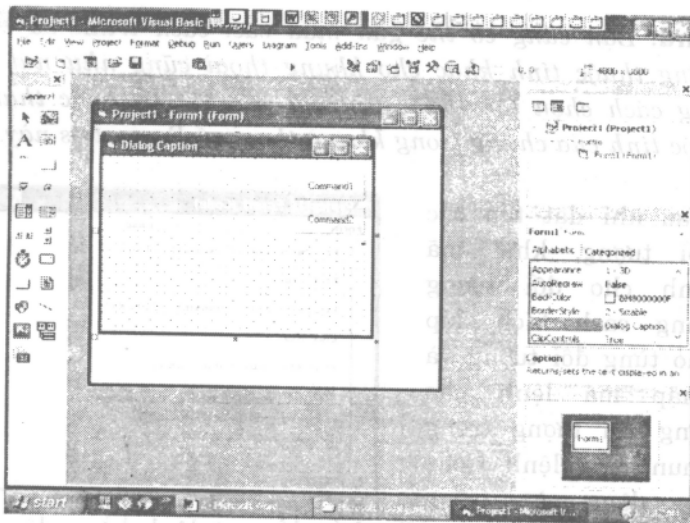


6. Bây giờ đặt tên cho khung thoại là Dialog Caption và các nút lệnh là OK và Cancel bằng cách:

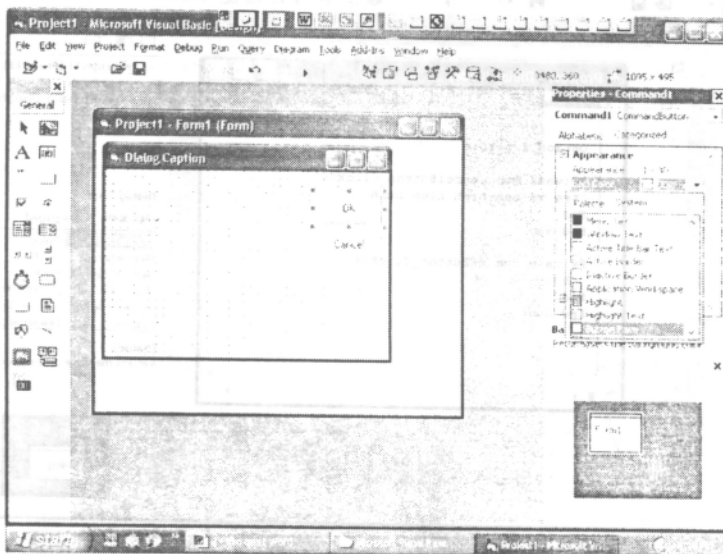
- Kích chọn khung thiết kế. Để ý có tám điểm chỉnh bằ ngay các cạnh và góc khung thiết kế.
- Trong khung cửa sổ Properties nằm bên phải khung cửa sổ thiết kế, kích thước lệnh Alphabetic hoặc Catagorized.
- Trong cột bên trái, kích Font, kích biểu tượng xỏ nằm cạnh tên phong mặc định trong cột bên phải và chọn loại phông cùng thuộc tính cho ký tự trong khung danh mục Font. Kích OK.



- Cũng trong phần Properties, kéo thanh trượt tìm và chọn thành phần Caption nằm trong cột bên trái. Trong cột bên phải, xóa tên mặc định và từ bàn phím, đặt tên cho khung thoại. Thí dụ, Dialog Caption. Lưu ý là trong khi đặt tên, từng ký tự sẽ hiện trong thanh tiêu đề của khung thiết kế.

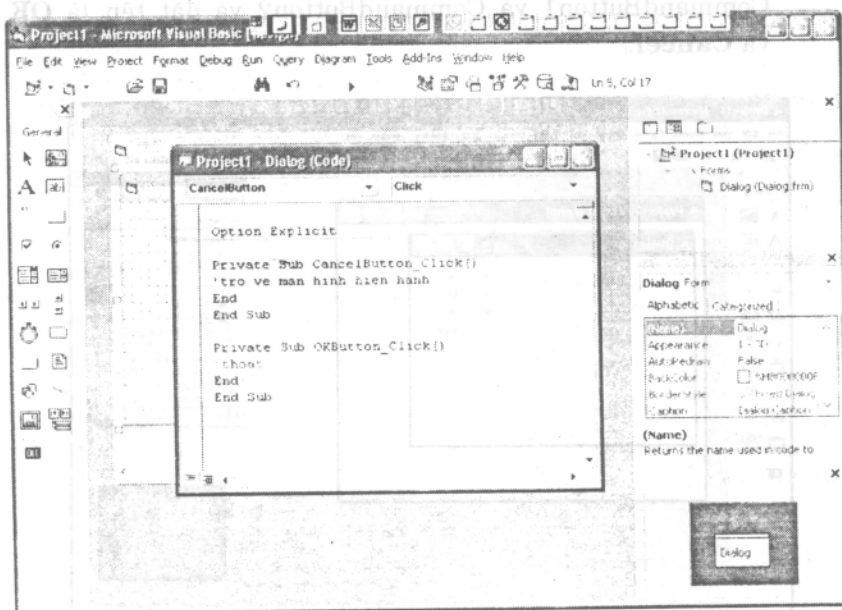
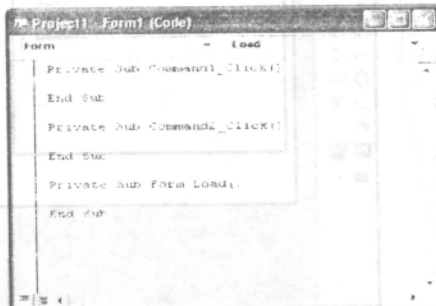


- e. Lần lượt áp dụng từ bước a đến d cho hai nút lệnh CommandButton1 và CommandButton2 và đặt tên là OK và Cancel.

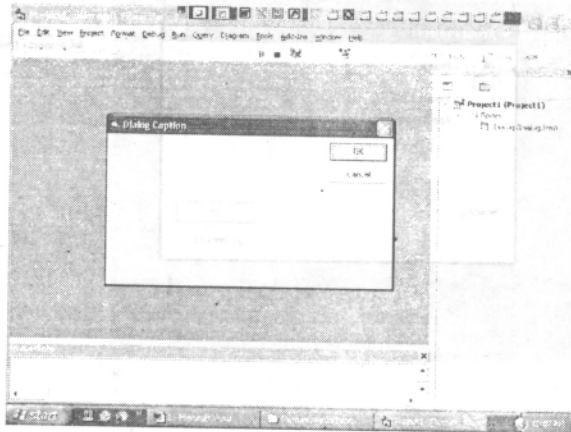


Ghi chú: Bạn cũng có thể gán màu sắc, cách trình bày cùng những thuộc tính khác cho khung thoại cũng như nút lệnh bằng cách chọn tên thuộc tính và gán giá trị hoặc tham số thuộc tính của chúng trong khung tham số Properties này.

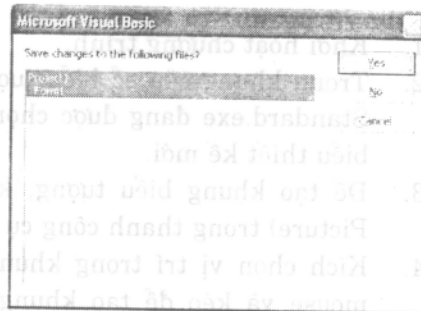
7. Sau khi đặt tên các đối tượng, khai mã lệnh cho đối tượng bằng cách kích đúp vào từng đối tượng và nhập mã lệnh cho từng đối tượng trong khung mã lệnh Code vào dòng còn trống (ngay vị trí con trỏ). Nhớ bắt đầu mã lệnh bằng dấu nhảy đơn và kết thúc mã lệnh và từ khóa End.



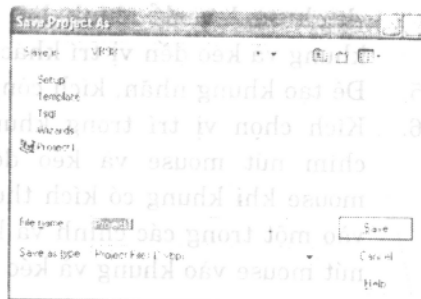
8. Để kiểm tra, từ menu Run, kích chọn Start hoặc ấn phím F5. Để trở về màn hình thiết kế, từ menu Run, chọn Stop hoặc kích công cụ Stop trong thanh công cụ.



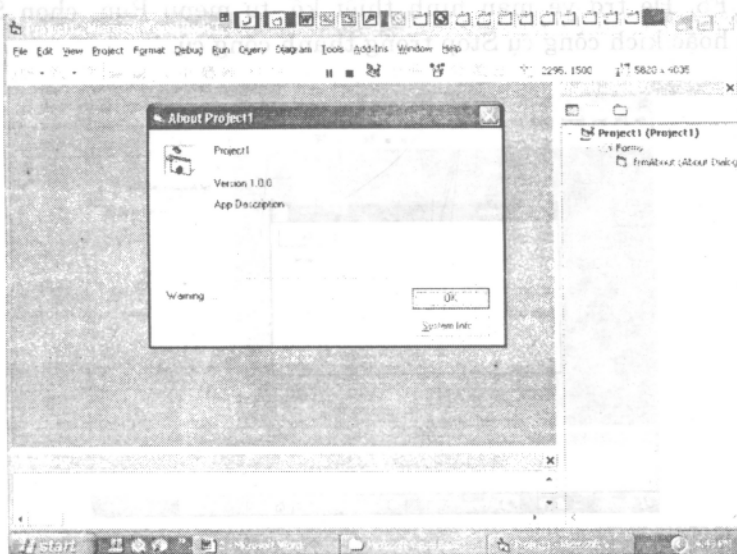
9. Sau khi kết thúc, để lưu lại bản thiết kế, từ menu File, chọn Save (As). Trong khung thoại yêu cầu xác nhận, kích Yes để lưu (hoặc No để thoát và không lưu lại, hoặc Cancel để hủy bỏ).



10. Trong khung thoại Save Project As, chọn thư mục và đặt tên cho bản thiết kế và kích Save.



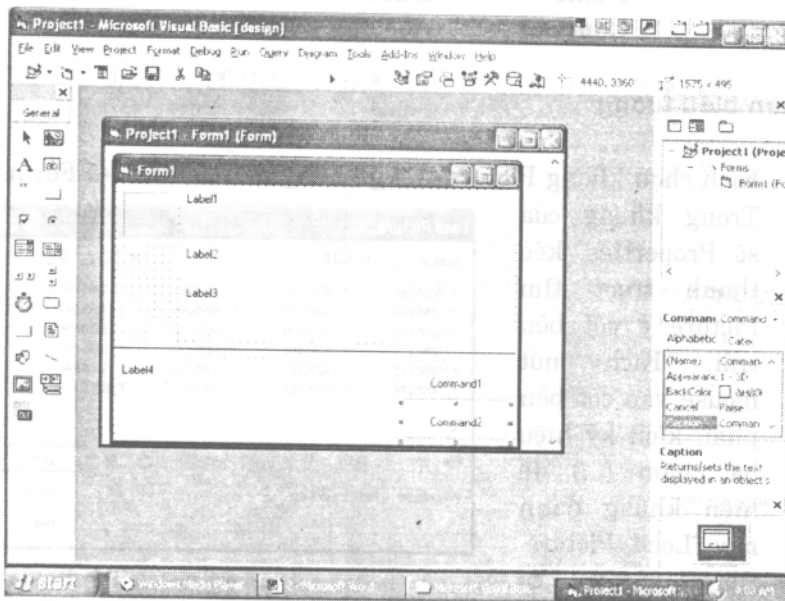
Bài 2: Tạo tập tin About Project1



1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Để tạo khung biểu tượng, kích công cụ Icon (hoặc  Picture) trong thanh công cụ General.
4. Kích chọn vị trí trong khung mẫu biểu Form, ấn chìm nút mouse và kéo để tạo khung. Thả nút mouse khi khung có kích thước đã chọn. Nếu cần thiết, kích vào một trong các chỉnh và kéo để chỉnh lại, hoặc kích chìm nút mouse vào khung và kéo đến vị trí khác.
5. Để tạo khung nhãn, kích công cụ Label trong thanh công cụ.
6. Kích chọn vị trí trong khung mẫu biểu Form, ấn  chìm nút mouse và kéo để tạo khung. Thả nút mouse khi khung có kích thước đã chọn. Nếu cần thiết, kích vào một trong các chỉnh và kéo để chỉnh lại, hoặc kích chìm nút mouse vào khung và kéo để chỉnh vị trí khung.

7. Thực hiện theo bước 5 và 6 để kẻ thêm ba khung nhân còn lại.
8. Để tạo đường phân giữa ba khung đầu tiên, kích công cụ Line trong thanh công cụ.
9. Kích chìm nút mouse vào khung mẫu biểu ngay vị trí đã chọn và kéo. Thả nút mouse khi đã có đường kẻ.
10. Để tạo hai nút lệnh, kích công cụ Command trong thanh công cụ General.
11. Kích chọn vị trí trong khung mẫu biểu Form, ấn chìm nút mouse và kéo để tạo nút lệnh. Thả nút mouse khi khung có kích thước đã chọn. Nếu cần thiết, kích vào một trong các chỉnh và kéo để chỉnh lại, hoặc kích chìm nút mouse vào khung và kéo để chỉnh vị trí khung.
12. Lặp lại bước 10 và 11 để tạo nút lệnh còn lại.

Đến đây, mẫu biểu thiết kế của bạn có dạng như sau:



Bây giờ, bạn bắt đầu gán thuộc tính cho các đối tượng.

Gán thuộc tính mẫu biểu

1. Kích chọn mẫu biểu.
2. Trong khung của sổ Properties, kéo thanh trượt tìm Font ở cột bên trái. Kích nút mouse vào cột bên phải, kích ký hiệu ba chấm (...) để hiện khung thuộc tính Font. Chọn loại phong và thuộc tính muốn áp dụng và kích OK.
3. Đặt tên và gán tiếp các thuộc tính cho khung Form như sau:

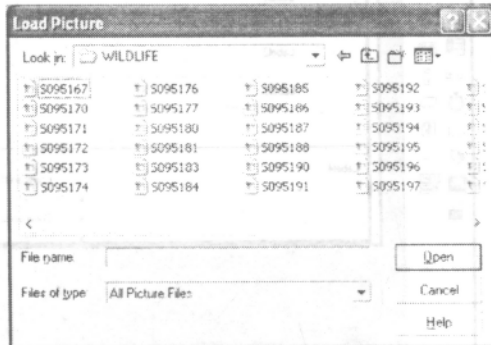
Name	frmAbout
Caption	About MyApp



Chèn biểu tượng

1. Kích chọn khung Picture trong khung của sổ thiết kế Form.

2. Trong khung của sổ Properties, kéo thanh trượt tìm Picture ở cột bên trái. Kích nút mouse vào cột bên phải, kích ký hiệu ba chấm (...) để hiện khung danh mục Load Picture. Chọn vị trí có chứa thư mục hình ảnh và chọn tập tin hình ảnh muốn tạo biểu tượng. Kích OK để tạo biểu tượng vào khung.



3. Cũng với khung cửa sổ thuộc tính, đặt tên và gán tiếp các thuộc tính cho khung biểu tượng như sau:

Name	petIcon
BackColor	chọn màu xám
BorderStyle	chọn màu xám đậm

Gán nhãn

1. Kích chọn khung nhãn Label1 trong mẫu biểu Form.
2. Trong khung cửa sổ thuộc tính Properties, lần lượt gán các thuộc tính cho nhãn Label1 như sau:

Font	MS Sans serif
Name	lblTitle
Caption	Application Title

3. Lặp lại từ bước 1 để tiếp tục gán nhãn cho các khung Label còn lại.

Đặt tên nút lệnh

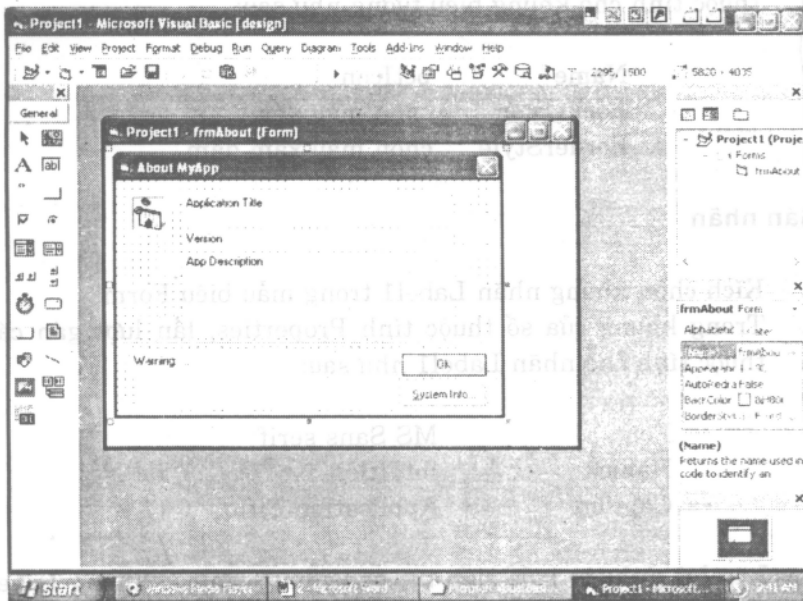
1. Kích chọn khung nhãn Command1 trong mẫu biểu Form.
2. Trong khung cửa sổ thuộc tính Properties, lần lượt gán các thuộc tính cho nhãn Command1 như sau:

Font	MS Sans Serif
Name	cmdOK
Caption	OK
Appearance	1 - 3D

3. Lặp lại từ bước 1 để đặt tên cho nút lệnh Command2:

Name	cmdSystem Info
Caption	&System Info...

Và khung thiết kế có dạng như sau:



Sau khi đã thiết kế xong mẫu biểu, bạn bắt đầu lập mã chương trình cho mẫu biểu.

1. Kích đúp vào khung mẫu biểu để hiện khung cửa sổ Code.
2. Kích con trỏ mouse vào dòng trống bên dưới Form Load() và từ bàn phím, bắt đầu nhập mã chương trình cho khung mẫu biểu.



3. Sau khi kết thúc, lần lượt kích đúp vào từng đối tượng và nhập mã lệnh tiếp cho đến khi kết thúc.
4. Cuối cùng, mã lệnh của tập tin About MyApp có dạng như sau:

Option Explicit

' Reg Key Security Options...

Const READ_CONTROL = &H20000

Const KEY_QUERY_VALUE = &H1

Const KEY_SET_VALUE = &H2

Const KEY_CREATE_SUB_KEY = &H4

Const KEY_ENUMERATE_SUB_KEYS = &H8

Const KEY_NOTIFY = &H10

Const KEY_CREATE_LINK = &H20

Const KEY_ALL_ACCESS = KEY_QUERY_VALUE +
KEY_SET_VALUE +
KEY_CREATE_SUB_KEY +
KEY_ENUMERATE_SUB_KEYS +
KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL

' Reg Key ROOT Types...

Const HKEY_LOCAL_MACHINE = &H80000002

Const ERROR_SUCCESS = 0

Const REG_SZ = 1 ' Unicode nul terminated string

Const REG_DWORD = 4 ' 32-bit number

Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared
Tools Location"

Const gREGVALSYSINFOLOC = "MSINFO"

Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared
Tools\MSINFO"

Const gREGVALSYSINFO = "PATH"

```
Private Declare Function RegOpenKeyEx Lib "advapi32" Alias  
"RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As  
String, ByVal ulOptions As Long, ByVal samDesired As Long,  
ByRef phkResult As Long) As Long
```

```
Private Declare Function RegQueryValueEx Lib "advapi32" Alias  
"RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName  
As String, ByVal lpReserved As Long, ByRef lpType As Long,  
ByVal lpData As String, ByRef lpcbData As Long) As Long
```

```
Private Declare Function RegCloseKey Lib "advapi32" (ByVal  
hKey As Long) As Long
```

```
Private Sub cmdSysInfo_Click()  
    Call StartSysInfo  
End Sub
```

```
Private Sub cmdOK_Click()  
    Unload Me  
End Sub
```

```
Private Sub Form_Load()  
    Me.Caption = "About " & App.Title  
    lblVersion.Caption = "Version " & App.Major & "." & App.Minor &  
        "." & App.Revision  
    lblTitle.Caption = App.Title  
End Sub
```

```
Public Sub StartSysInfo()  
    On Error GoTo SysInfoErr  
  
    Dim rc As Long  
    Dim SysInfoPath As String
```

... Try To Get System Info Program Path\Name From Registry...

```

If
    GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
                gREGVALSYSINFO, SysInfoPath) Then
        ' Try To Get System Info Program Path Only From
        Registry...
ElseIf
    GetKeyValue(HKEY_LOCAL_MACHINE,
                gREGKEYSYSINFOLOC,      gREGVALSYSINFOLOC,
                SysInfoPath) Then
        ' Validate Existence Of Known 32 Bit File Version
        If (Dir(SysInfoPath & "MSINFO32.EXE") <> "") Then
            SysInfoPath = SysInfoPath & "MSINFO32.EXE"
            ' Error - File Can Not Be Found...
        Else
            GoTo SysInfoErr
        End If
    ' Error - Registry Entry Can Not Be Found...
Else
    GoTo SysInfoErr
End If

Call Shell(SysInfoPath, vbNormalFocus)

Exit Sub
SysInfoErr:
    MsgBox "System Information Is Unavailable At This Time",
        vbOKOnly
End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As
String, SubKeyRef As String, ByRef KeyVal As String) As Boolean
    Dim i As Long                ' Loop Counter
    Dim rc As Long                ' Return Code
    Dim hKey As Long              ' Handle To An Open Registry Key
    Dim hDepth As Long

```

```

Dim KeyValType As Long      ' Data Type Of A Registry Key
Dim tmpVal As String        ' Tempory Storage For A Registry Key
                              Value
Dim KeyValSize As Long      ' Size Of Registry Key Variable
'-----
' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
'-----
rc      =      RegOpenKeyEx(KeyRoot,      KeyName,      0,
                              KEY_ALL_ACCESS, hKey)
      ' Open Registry Key

If (rc <> ERROR_SUCCESS) Then
      GoTo GetKeyError ' Handle Error...

tmpVal = String$(1024, 0)      ' Allocate Variable Space
KeyValSize = 1024              ' Mark Variable Size

'-----
' Retrieve Registry Key Value...
'-----
rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
      KeyValType, tmpVal, KeyValSize) ' Get/Create Key Value

If (rc <> ERROR_SUCCESS) Then
      GoTo GetKeyError ' Handle Errors

If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then
      ' Win95 Adds Null Terminated String...
      tmpVal = Left(tmpVal, KeyValSize - 1) ' Null Found, Extract
                                          From String
Else      ' WinNT Does NOT Null Terminate String...
      tmpVal = Left(tmpVal, KeyValSize) ' Null Not Found,
                                          Extract String Only
End If

```



```

'-----
' Determine Key Value Type For Conversion...
'-----

```

```

Select Case KeyValType      ' Search Data Types...
Case REG_SZ                ' String Registry Key Data Type
    KeyVal = tmpVal        ' Copy String Value
Case REG_DWORD             ' Double Word Registry Key Data Type
    For i = Len(tmpVal) To 1 Step -1    ' Convert Each Bit
        KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1)))
    ' Build Value Char. By Char.
    Next
    KeyVal = Format$("&h" + KeyVal) ' Convert Double Word To
                                   String
End Select

```

```

GetKeyValue = True          ' Return Success
rc = RegCloseKey(hKey)     ' Close Registry Key
Exit Function              ' Exit

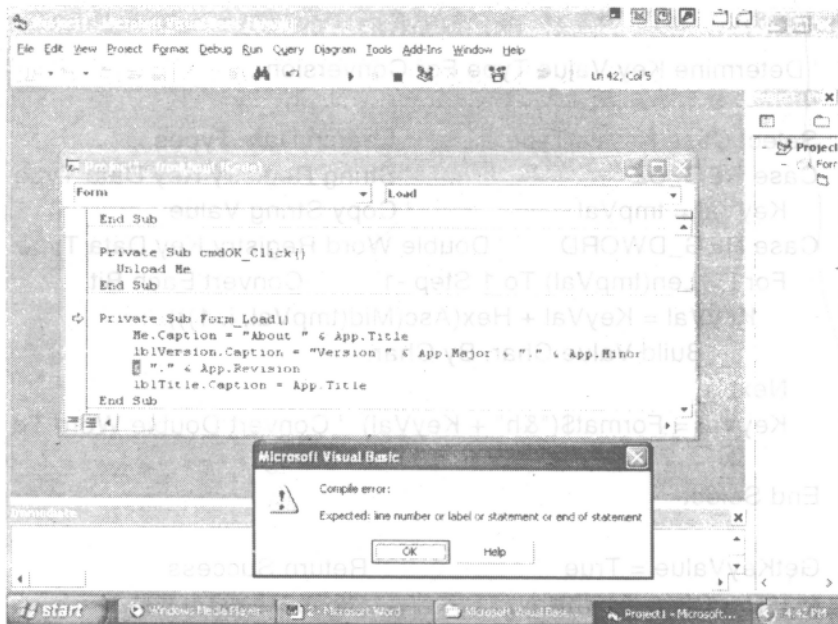
```

```

GetKeyError:    ' Cleanup After An Error Has Occured...
KeyVal = ""     ' Set Return Val To Empty String
GetKeyValue = False    ' Return Failure
rc = RegCloseKey(hKey) ' Close Registry Key
End Function

```

- Để kiểm tra lỗi chương trình, kích menu Debug và kích Step Into. Nếu phát hiện lỗi, dòng lỗi sẽ được phủ màu vàng với mũi tên xác định bên trái. Cùng lúc, đối tượng lỗi sẽ được phủ màu xanh và câu lệnh lỗi hiện màu đỏ và đồng thời, màn hình hiện khung thoại báo lỗi cùng với thông tin đề nghị sửa lỗi. Trong thí dụ này, mã lệnh của mẫu biểu Form Load() bị lỗi và toán tử & cùng với biểu thức bị lỗi do biểu thức này đã xuống dòng không đúng vị trí. Khắc phục lỗi và lần lượt tìm các lỗi khác (nếu có) trong chương trình.

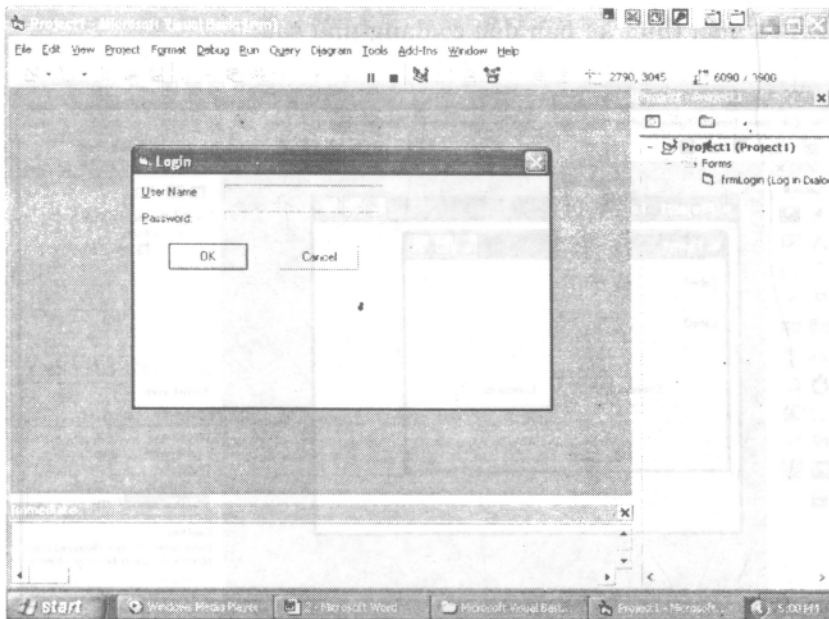


6. Ấn phím F5 để điều hoạt chương trình.
7. Sau khi kết thúc, dùng lệnh Save As từ menu File để lưu tập tin với tên mới.

Bài 3: Lập trình khung nhập mật mã Login

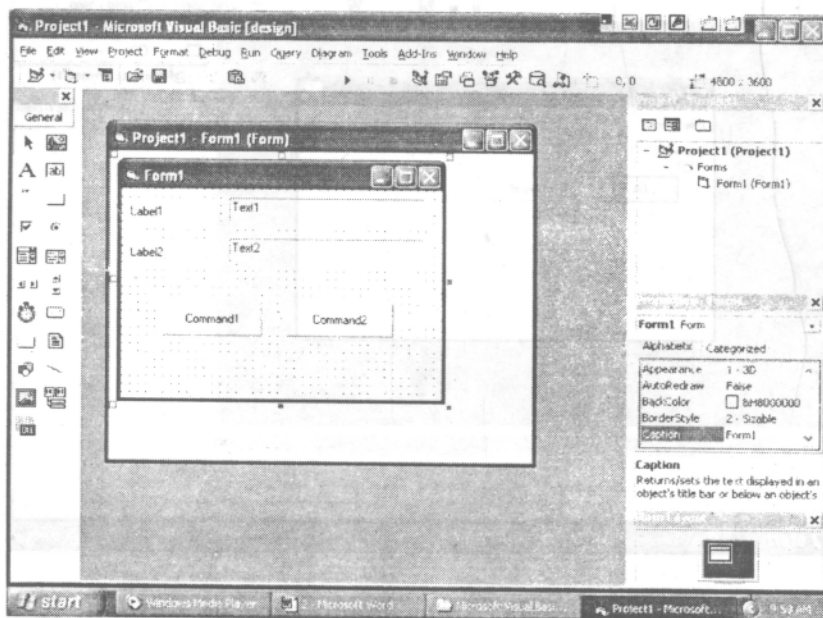
Trong bài tập này, bạn thiết kế một khung nhập mật mã gồm khung nhập tên người sử dụng, khung nhập mật mã và hai nút lệnh. Nếu mật mã nhập sai và kích OK, màn hình hiện khung thoại báo lỗi.

1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.



3. Kích biểu tượng Label trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung cho nhãn Label1. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp hoặc kích và kéo để định lại vị trí.
4. Kích công cụ TextBox trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung cho nhãn Text1. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp hoặc kích và kéo để định lại vị trí.
5. Lặp lại các bước 3 và 4 để tạo hai khung Label2 và Text2.
6. Kích công cụ Command trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung cho nhãn nút lệnh Command1. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp hoặc kích và kéo để định lại vị trí.
7. Lặp lại bước 6 để tạo khung nút lệnh Command2.

Và mẫu thiết kế ban đầu có dạng như sau:



Đến đây bắt đầu đặt tên mẫu biểu, gán nhãn và thuộc tính cho khung nhập mật mã.

Đặt tên mẫu biểu

1. Kích chọn mẫu biểu.
2. Trong khung cửa sổ Properties, đặt tên và gán thuộc tính cho khung Form1 như sau:

Font MS Sans Serif
Name FrmLogin

Gán nhãn

1. Kích chọn khung nhãn Label1.

2. Trong khung cửa sổ Properties, đặt tên và gán thuộc tính cho khung Label1 như sau:

Font	MS Sans Serif
Name	lblLabels
Alignment	Left
Caption	&Username:

3. Kích chọn khung nhãn Label2.
4. Trong khung cửa sổ Properties, đặt tên và gán thuộc tính cho khung Label2 như sau:

Font	MS Sans Serif
Name	lblLabels
Alignment	Left
Caption	&Password:

Thay đổi tham số khung ký tự Text

1. Kích chọn khung ký tự Text1.
2. Trong khung cửa sổ Properties, thay đổi thuộc tính cho khung ký tự Text1 như sau:

Font	MS Sans Serif
Name	txtUserName
Alignment	Left
Text	xóa ký tự Text1 tạo khung ký tự trống để người sử dụng tự nhập tên của họ.

3. Kích chọn khung ký tự Text2 để thay đổi thuộc tính cho khung ký tự Password và khi người sử dụng nhập ký tự bất kỳ vào khung Password sẽ hiện những dấu hoa thị.

Font	MS Sans Serif
Name	txtPassword

Alignment	Left
Text	xóa ký tự Text2 tạo khung ký tự trắng để người sử dụng tự nhập mật mã.
PasswordChar	*

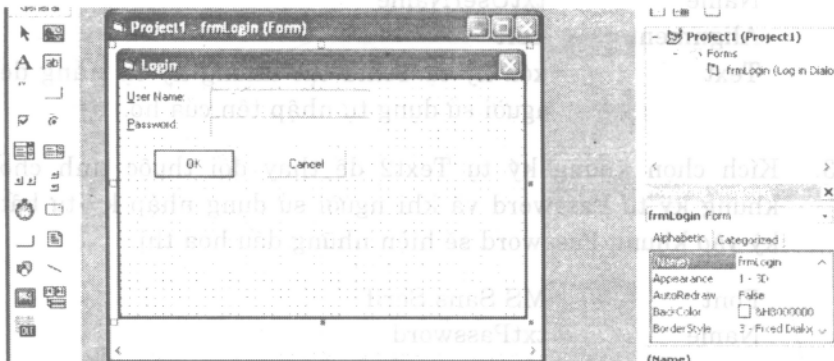
Gán thuộc tính nút lệnh

1. Kích chọn nút lệnh Command1.
2. Trong khung thuộc tính Properties, thay đổi và gán thuộc tính nút lệnh như sau:

Font	MS Sans Serif
Name	cmdOK
Alignment	Left
Caption	OK
Cancel	False
Default	True

3. Gán thuộc tính cho nút lệnh Command2 như sau:

Font	MS Sans Serif
Name	cmdCancel
Alignment	Left
Caption	Cancel
Cancel	True
Default	False



Đến đây, lần lượt kích đúp vào từng đối tượng để mở khung cửa sổ Code và lập trình cho từng đối tượng như sau:

Option Explicit

Public LoginSucceeded As Boolean

Private Sub cmdCancel_Click()

 'gán biến chung về FALSE

 'để thông báo là mật mã nhập không đúng

 LoginSucceeded = False

 Me.Hide

End Sub

Private Sub cmdOK_Click()

 'Kiểm tra lại mật mã

 If txtPassword = "password" Then

 'gán mật mã để chuyển hành động sang câu lệnh gọi

 'gán tham biến chung là cách dễ nhất

 LoginSucceeded = True

 Me.Hide

 Else

 MsgBox "Invalid Password, try again!", , "Login"

 txtPassword.SetFocus

 SendKeys "{Home}+[End]"

 End If

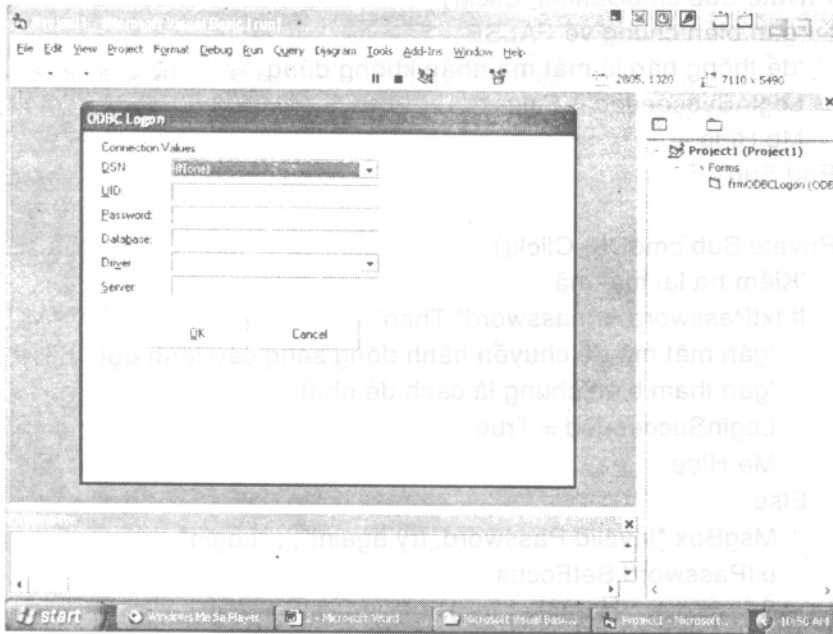
End Sub

Sau khi sửa lỗi với lệnh Debug, ấn F5 để điều hoạt và sau đó dùng lệnh File/Save As để lưu tập tin theo tên tùy chọn.

Bài 4: Thiết kế khung thoại ODBC Logon

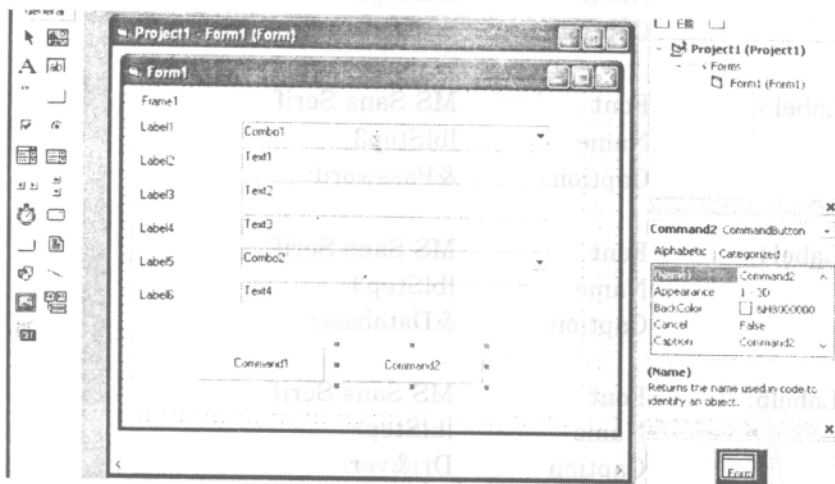
Theo hình minh họa trên, bạn thiết kế khung hòa mạng để người sử dụng khai báo những giá trị, tham số cần thiết khi cần

truy cập dữ liệu hoặc đối tượng từ máy chủ. Trong khung thoại này, có sáu khung nhãn: nhãn DSN chứa các tập tin dữ liệu, khi kích mũi tên xổ sẽ hiện danh mục các chương trình tập tin dữ liệu, một nhãn chứa các đường dẫn vào tập tin dữ liệu. Bốn nhãn còn lại gồm một nhãn khai báo mật mã, một nhãn khai báo tên máy chủ, một nhãn khai báo hệ cơ sở dữ liệu và một nhãn khai báo số nhận dạng. Hai nút lệnh là OK và Cancel.



1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích biểu tượng Frame trong thanh công cụ General. xv
Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp.

4. Kích biểu tượng Label trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung cho nhãn Label1. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp hoặc kích và kéo để định lại vị trí.
5. Lặp lại bước 4 để tạo khung các nhãn còn lại.
6. Để tạo danh sách xổ các tập tin dữ liệu chứa trong chương trình để người sử dụng kích chọn, kích công cụ ComboBox trong thanh công cụ General. Kích vị trí nằm cạnh khung nhãn Label1 và kéo để tạo khung. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp.
7. Áp dụng thủ tục này cho khung nhãn Label5.
8. Để tạo khung nhập các thông tin cho các nhãn còn lại, kích công cụ TextBox trong thanh công cụ General. Kích chọn vị trí nằm cạnh nhãn Label2 và kéo để tạo khung ký tự. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp.
9. Thực hiện lại bước 8 để tạo khung ký tự cho các nhãn còn lại.
10. Kích công cụ CommandButton và lần lượt tạo hai nút lệnh Command1 và Command2.



Đến đây, bạn đặt tên và gán thuộc tính cho mẫu biểu và các đối tượng qua cách dùng khung cửa sổ thuộc tính Properties/Categorized.

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	frmODBCLogon
Caption	ODBC Logon

Thay đổi tên nhãn

Frame1:	Font	MS Sans Serif
	Name	fraStep3
	Caption	Connection Value

Label1:	Font	MS Sans Serif
	Name	lblStep3
	Caption	&DSN:

Label2:	Font	MS Sans Serif
	Name	lblStep3
	Caption	&UID:

Label3:	Font	MS Sans Serif
	Name	lblStep3
	Caption	&Password:

Label4:	Font	MS Sans Serif
	Name	lblStep3
	Caption	&Database:

Label5:	Font	MS Sans Serif
	Name	lblStep3
	Caption	Dri&ver:

Label6:	Font Name Caption	MS Sans Serif lblStep3 &Server:
---------	-------------------------	---------------------------------------

Thay đổi tên khung ký tự

Text1	Font Name	MS Sans Serif txtUID
-------	--------------	-------------------------

Text2	Font Name	MS Sans Serif txtPWD
-------	--------------	-------------------------

Text3	Font Name	MS Sans Serif txtDatabase
-------	--------------	------------------------------

Text4	Font Name	MS Sans Serif txtServer
-------	--------------	----------------------------

Combo1	Font Name ItemData List Style Text	MS Sans Serif cboDSNList (List) (List) 2- DropDownList cboDSNList
--------	---	--

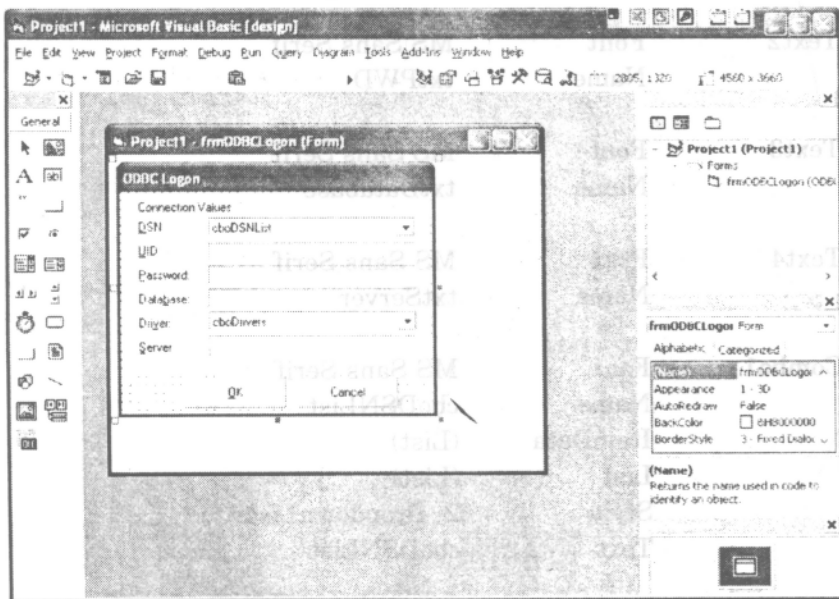
Combo2	Font Name ItemData List Style Text	MS Sans Serif cboDrivers (List) (List) 2- DropDownList cboDrivers
--------	---	--

Thay đổi thuộc tính nút lệnh

Command1	Font	MS Sans Serif
----------	------	---------------

	Name	cmdOK
	Caption	&OK
Command2	Font	MS Sans Serif
	Name	cmdCancel
	Caption	&Cancel

Cuối cùng mẫu biểu của bạn có dạng như sau:



Lần lượt kích đúp vào từng đối tượng mà nhập mã lệnh trong khung cửa sổ Code như sau:

Option Explicit

```
Private Declare Function SQLDataSources Lib "ODBC32.DLL"
    (ByVal henv&, ByVal fDirection%, ByVal szDSN$, ByVal
    cbDSNMax%, pcbDSN%, ByVal szDescription$, ByVal
    cbDescriptionMax%, pcbDescription%) As Integer
```

```

Private Declare Function SQLAllocEnv% Lib "ODBC32.DLL"
(env&)
Const SQL_SUCCESS As Long = 0
Const SQL_FETCH_NEXT As Long = 1

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim sConnect As String
    Dim sADOConnect As String
    Dim sDAOConnect As String
    Dim sDSN As String

    If cboDSNList.ListIndex > 0 Then
        sDSN = "DSN=" & cboDSNList.Text & ";"
    Else
        sConnect = sConnect & "Driver=" & cboDrivers.Text & ";"
        sConnect = sConnect & "Server=" & txtServer.Text & ";"

    End If

    sConnect = sConnect & "UID=" & txtUID.Text & ";"
    sConnect = sConnect & "PWD=" & txtPWD.Text & ";"

    If Len(txtDatabase.Text) > 0 Then
        sConnect = sConnect & "Database=" & txtDatabase.Text &
        ";"
    End If

    sADOConnect = "PROVIDER=MSDASQL;" & sDSN &
sConnect
    sDAOConnect = "ODBC;" & sDSN & sConnect

    MsgBox _

```

```

        "To open an ADO Connection, use:" & vbCrLf & _
        "Set gConnection = New Connection" & vbCrLf & _
        "gConnection.Open "" & sADODConnect & "" & vbCrLf &
vbCrLf & _
        "To open a DAO database object. use:" & vbCrLf & _
        "Set gDatabase = OpenDatabase(vbNullString, 0, 0,
                                sADODConnect)" & vbCrLf & vbCrLf & _
        "Or to open an RDO Connection, use:" & vbCrLf & _
        "Set gRDOConnection =
                                rdoEnvironments(0).OpenConnection(sD
                                SN, rdDriverNoPrompt, 0, sConnect)"

'ADO:
'Set gConnection = New Connection
'gConnection.Open sADODConnect
'DAO:
'Set gDatabase = OpenDatabase(vbNullString, 0, 0,
sADODConnect)
'RDO:
'Set gRDOConnection =
                                rdoEnvironments(0).OpenConnection(s
                                DSN, rdDriverNoPrompt, 0, sConnect)

End Sub
Private Sub Form_Load()
    GetDSNsAndDrivers
End Sub

Private Sub cboDSNList_Click()
    On Error Resume Next
    If cboDSNList.Text = "(None)" Then
        txtServer.Enabled = True
        cboDrivers.Enabled = True
    Else
        txtServer.Enabled = False
        cboDrivers.Enabled = False
    End If
End Sub

```

```

End If
End Sub

Sub GetDSNsAndDrivers()
    Dim i As Integer
    Dim sDSNItem As String * 1024
    Dim sDRVItem As String * 1024
    Dim sDSN As String
    Dim sDRV As String
    Dim iDSNLen As Integer
    Dim iDRVLen As Integer
    Dim lHenv As Long      'handle to the environment

    On Error Resume Next
    cboDSNList.AddItem "(None)"

    'get the DSNs
    If SQLAllocEnv(lHenv) <> -1 Then
        Do Until i <> SQL_SUCCESS
            sDSNItem = Space$(1024)
            sDRVItem = Space$(1024)
            i = SQLDataSources(lHenv, SQL_FETCH_NEXT,
                               sDSNItem, 1024, iDSNLen, sDRVItem, 1024,
                               iDRVLen)
            sDSN = Left$(sDSNItem, iDSNLen)
            sDRV = Left$(sDRVItem, iDRVLen)
            If sDSN <> Space(iDSNLen) Then
                cboDSNList.AddItem sDSN
                cboDrivers.AddItem sDRV
            End If
        Loop
    End If

    'remove the dupes
    If cboDSNList.ListCount > 0 Then
        With cboDrivers

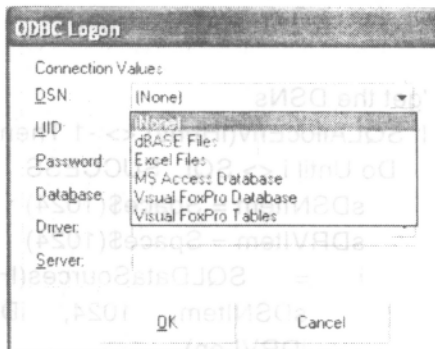
```

```

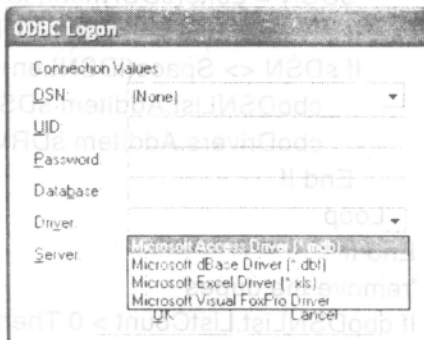
If .ListCount > 1 Then
    i = 0
    While i < .ListCount
        If .List(i) = .List(i + 1) Then
            .RemoveItem (i)
        Else
            i = i + 1
        End If
    Wend.
End If
End With
End If
cboDSNList.ListIndex = 0
End Sub

```

Sau khi nhập mã lệnh và sửa lỗi, ấn phím F5 để điều hoạt chương trình. Để kiểm tra, trong khung màn hình điều hoạt, kích mũi tên xổ của đối tượng DSN và thành phần này sẽ hiện các tập tin chương trình.

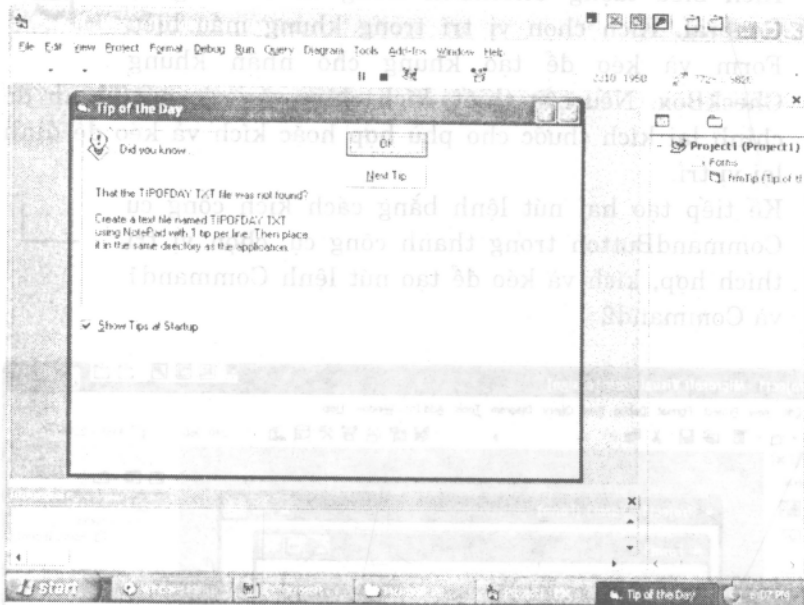


Và cũng tương tự đối với Driver.



Bài 5: Tạo tập tin Tip Of The Day

Chúng ta thường để ý là sau khi cài đặt và khởi động chương trình, màn hình hiện khung thoại Tip of the Day và chúng ta có thể cho hiện hoặc giấu bằng cách kích nút chọn Show như hình minh họa. Nhưng làm cách nào chúng ta có thể thiết kế được?

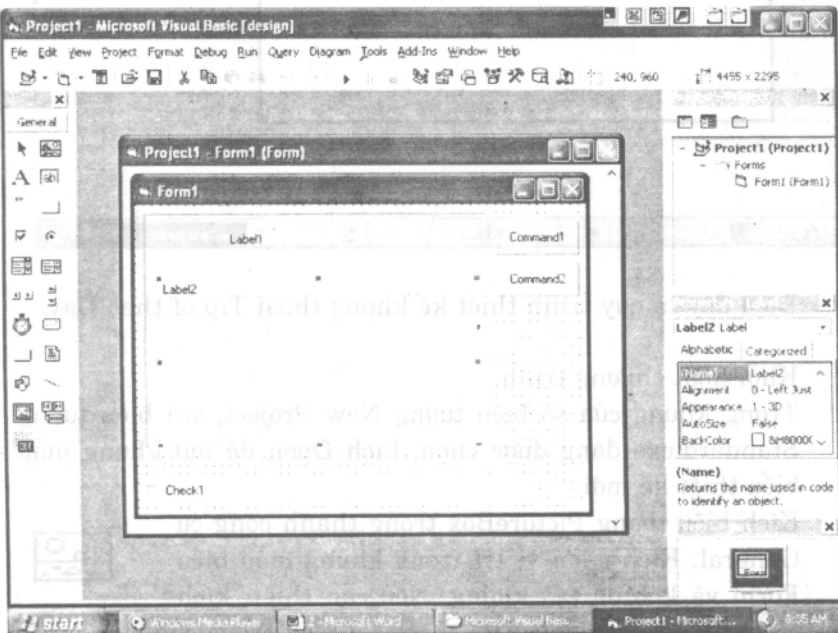


Dưới đây là quy trình thiết kế khung thoại Tip of the Day.

1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích biểu tượng PictureBox trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp.



4. Kích biểu tượng Label trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung cho nhãn Label1. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp hoặc kích và kéo để định lại vị trí.
5. Lặp lại bước 4 để tạo khung nhãn cho Label2.
6. Kích biểu tượng CheckBox trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung cho nhãn khung CheckBox. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp hoặc kích và kéo để định lại vị trí.
7. Kế tiếp tạo hai nút lệnh bằng cách kích công cụ CommandButton trong thanh công cụ, chọn vị trí thích hợp, kích và kéo để tạo nút lệnh Command1 và Command2.



Đến đây, bạn lần lượt gán thuộc tính cho các đối tượng.

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	frmTip
Caption	Tip of the Day

Gán thuộc tính khung Picture

Font	MS Sans Serif
Name	Picture1
BackColor	màu trắng
Picture	Icon
ScaleMode	1 – Twip

Gán thuộc tính khung Label1

Font	MS Sans Serif
Name	Label1
Alignment	Left
BackColor	màu trắng
Caption	Did you know...

Gán thuộc tính khung Label2

Font	MS Sans Serif
Name	lblTipText
Alignment	Left
BackColor	màu trắng

Gán thuộc tính khung Checkbox

Font	MS Sans Serif
Name	chkLoadTipAtStartup

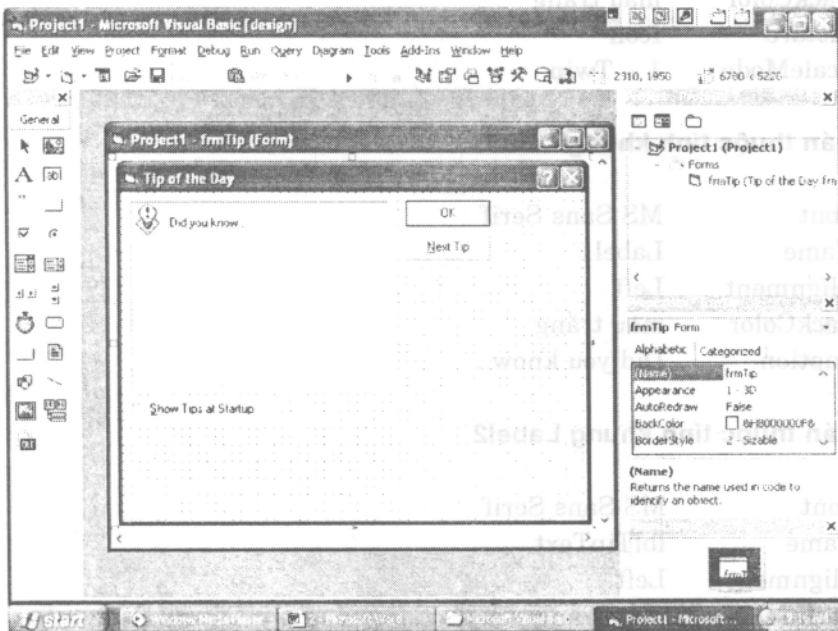
Alignment Left
Caption &Show Tips At Startup

Gán thuộc tính nút lệnh Command1

Name cmdOK
Caption OK

Gán thuộc tính nút lệnh Command2

Name cmdOK
Caption &Next Tip



Đến đây, lần lượt kích đúp vào từng đối tượng và nhập mã lệnh của chúng trong khung cửa sổ Code.

Option Explicit

' The in-memory database of tips.

Dim Tips As New Collection

' Name of tips file

Const TIP_FILE = "TIPOFDAY.TXT"

' Index in collection of tip currently being displayed.

Dim CurrentTip As Long

Private Sub DoNextTip()

 ' Select a tip at random.

 CurrentTip = Int((Tips.Count * Rnd) + 1)

 ' Or, you could cycle through the Tips in order

 ' CurrentTip = CurrentTip + 1

 ' If Tips.Count < CurrentTip Then

 ' CurrentTip = 1

 ' End If

 ' Show it.

 frmTip.DisplayCurrentTip

End Sub

Function LoadTips(sFile As String) As Boolean

 Dim NextTip As String ' Each tip read in from file.

 Dim InFile As Integer ' Descriptor for file.

 ' Obtain the next free file descriptor.

 InFile = FreeFile

```

' Make sure a file is specified.
If sFile = "" Then
    LoadTips = False
    Exit Function
End If
' Make sure the file exists before trying to open it.
If Dir(sFile) = "" Then
    LoadTips = False
    Exit Function
End If

' Read the collection from a text file.
Open sFile For Input As InFile
While Not EOF(InFile)
    Line Input #InFile, NextTip
    Tips.Add NextTip
Wend
Close InFile

' Display a tip at random.
DoNextTip

LoadTips = True

End Function

Private Sub chkLoadTipsAtStartup_Click()
' save whether or not this form should be displayed at startup
SaveSetting App.EXENAME, "Options", "Show Tips at Startup",
chkLoadTipsAtStartup.Value
End Sub

Private Sub cmdNextTip_Click()
DoNextTip
End Sub

```

```
Private Sub cmdOK_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Dim ShowAtStartup As Long
```

```
    ' See if we should be shown at startup
```

```
    ShowAtStartup = GetSetting(App.EXENAME, "Options", "Show  
Tips at Startup", 1)
```

```
    If ShowAtStartup = 0 Then
```

```
        Unload Me
```

```
        Exit Sub
```

```
    End If
```

```
    ' Set the checkbox, this will force the value to be written back  
out to the registry
```

```
    Me.chkLoadTipsAtStartup.Value = vbChecked
```

```
    ' Seed Rnd
```

```
    Randomize
```

```
    ' Read in the tips file and display a tip at random.
```

```
    If LoadTips(App.Path & "\ " & TIP_FILE) = False Then
```

```
        lblTipText.Caption = "That the " & TIP_FILE & " file was not  
found? " & vbCrLf & vbCrLf & _
```

```
        "Create a text file named " & TIP_FILE & " using NotePad  
with 1 tip per line. " & _
```

```
        "Then place it in the same directory as the application. "
```

```
    End If
```

```
End Sub
```

```
Public Sub DisplayCurrentTip()
```

```
    If Tips.Count > 0 Then
```

```

lblTipText.Caption = Tips.Item(CurrentTip)
End If
End Sub

```

Sau khi kết thúc nhập mã lệnh chương trình trong khung của sổ Code, bắt đầu sửa lỗi với lệnh Debug và sau khi sửa lỗi, ấn phím F5 để điều hoạt.

Sau khi kết thúc, lưu lại với lệnh Save As và đặt tên tập tin cho mẫu biểu thiết kế.

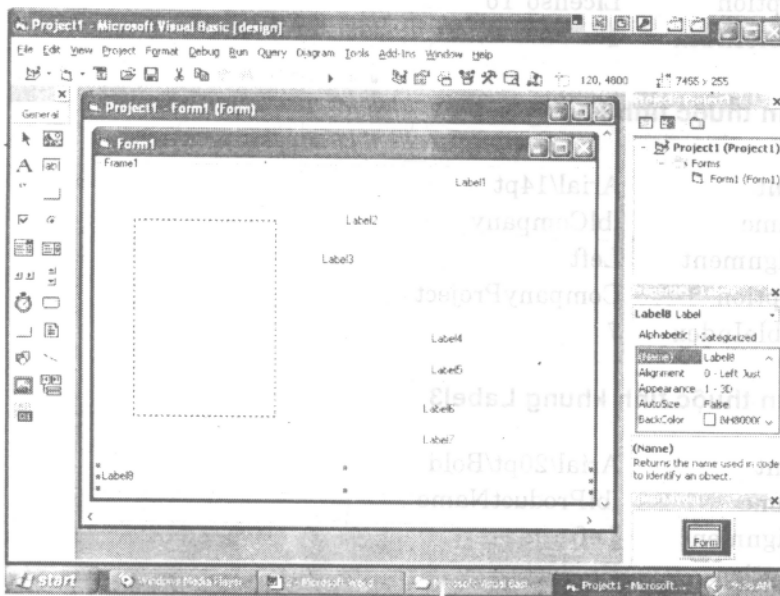
Bài 6: Tạo tập tin Giới thiệu sản phẩm

Với bài tập này, bạn có thể thay đổi nội dung, tham số trong phần khai báo thuộc tính để lập trình khung giới thiệu sản phẩm theo ý muốn.



1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích công cụ FrameBox trong thanh công cụ General. Kích và kéo mouse trong khung cửa sổ form để tạo khung có kích thước theo ý muốn.
4. Kích biểu tượng ImageBox trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp.
5. Kích biểu tượng Label trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung cho nhãn Label1. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp hoặc kích và kéo để định lại vị trí.
6. Áp dụng bước 5 để tạo những khung Label còn lại.

xv



Đặt tên và gán các thuộc tính đối tượng

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	frmSplash
LinkTopic	Form2

Gán thuộc tính khung ImageBox

Name	imgLogo
Picture	Icon

Gán thuộc tính khung Label1

Font	Arial/8pt
Name	lblLicenseTo
Alignment	Right
Caption	Licenso To
TabIndex	1

Gán thuộc tính khung Label2

Font	Arial/14pt
Name	lblCompany
Alignment	Left
Caption	CompanyProject
TabIndex	7

Gán thuộc tính khung Label3

Font	Arial/20pt/Bold
Name	lblProductName
Alignment	Left
Caption	Product
TabIndex	8

Gán thuộc tính khung Label4

Font	Arial/14pt/Bold
Name	lblPlatform
Alignment	Right
Caption	Platform
TableIndex	6

Gán thuộc tính khung Label5

Font	Arial/10pt
Name	lblVersion
Alignment	Right
Caption	Version
TableIndex	5

Gán thuộc tính khung Label6

Font	Arial/8pt
Name	lblCopyright
Alignment	Left
Caption	Copyright
TableIndex	4

Gán thuộc tính khung Label7

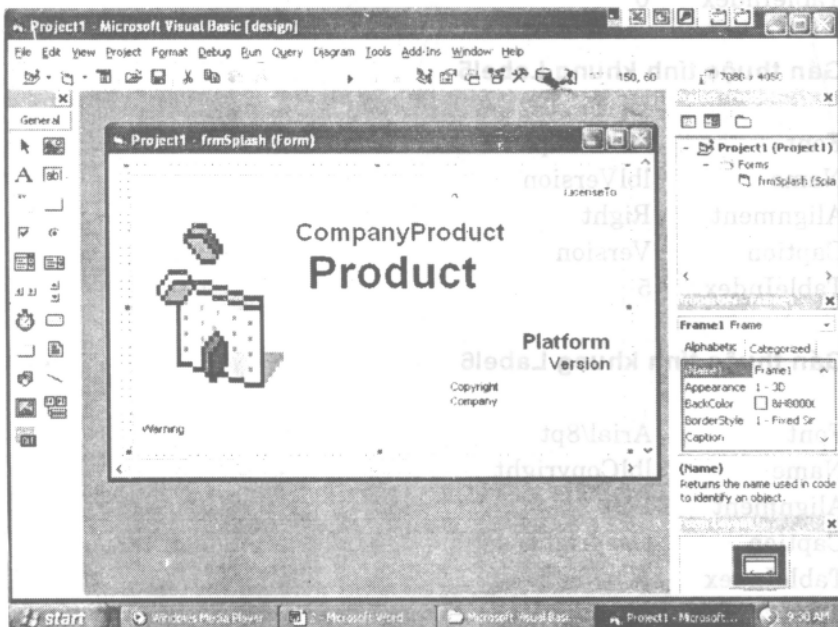
Font	Arial/8pt
Name	lblCompany
Alignment	Left
Caption	Copyright
TableIndex	3

Gán thuộc tính khung Label6

Font	Arial/8pt
------	-----------

Name	lblWarning
Alignment	Left
Caption	Warning
TabIndex	2

Mẫu biểu thiết kế lúc này có dạng như sau:



Lần lượt kích đúp vào từng đối tượng và nhập chương trình cho chúng trong khung cửa sổ Code.

Option Explicit

Private Sub Form_KeyPress(KeyAscii As Integer)

Unload Me

End Sub

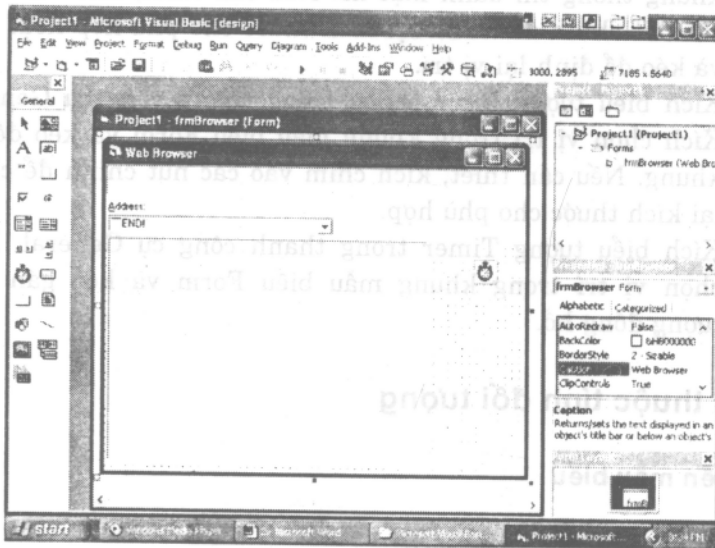
```

Private Sub Form_Load()
    lblVersion.Caption = "Version " & App.Major & "." & App.Minor &
        "." & App.Revision
    lblProductName.Caption = App.Title
End Sub

Private Sub Frame1_Click()
    Unload Me
End Sub

```

Bài 7: Tạo khung cửa sổ trình duyệt Web



1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích công cụ FrameBox trong thanh công cụ General. Kích và kéo mouse trong khung cửa sổ form để tạo khung có kích thước theo ý muốn.

4. Kích biểu tượng Toolbox trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp.
5. Kích biểu tượng Label trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung cho nhãn Label1. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp hoặc kích và kéo để định lại vị trí.
6. Kích biểu tượng ComboBox trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung thông tin danh mục xổ. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp hoặc kích và kéo để định lại vị trí.
7. Kích biểu tượng BrowserBox trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo để tạo khung. Nếu cần thiết, kích chìm vào các nút chỉnh để chỉnh lại kích thước cho phù hợp.
8. Kích biểu tượng Timer trong thanh công cụ General. Kích chọn vị trí trong khung mẫu biểu Form và kéo gắn biểu tượng đồng hồ.

Gán thuộc tính đối tượng

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	frmBrowser
Caption	Web Browser

Gán thuộc tính khung Toolbox

Name	tbToolBar
Align	Top

Gán thuộc tính khung Label1

Name	tbToolbox
Alignment	Left
Caption	&Address
Tag	&Address

Gán thuộc tính khung Label1

Font	MS Sans Srif/8pt
Name	lblAddress
Alignment	Left
Caption	&Address
Tag	&Address

Gán thuộc tính khung ComboBox

Font	MS Sans Srif/8pt
Name	cboAddress
ItemData	(List)
List	(List)
Style	DropdownCombo
Text	— END!
TabIndex	2

Gán thuộc tính khung Browser

Font	MS Sans Srif/8pt
Name	brwWebBrowser
HasDC	True
Negotiate	False

Gán thuộc tính Timer

Name	timTimer
Interval	5

Sau khi thay đổi và nhập các thuộc tính cho những đối tượng trong khung thiết kế mẫu biểu Form, tuần tự kích đúp vào từng đối tượng và lập trình cho chúng trong khung cửa sổ Code như sau:

Option Explicit

Public StartingAddress As String

Dim mbDontNavigateNow As Boolean

Private Sub Form_Load()

On Error Resume Next

Me.Show

tbToolBar.Refresh

Form_Resize

cboAddress.Move 50, lblAddress.Top + lblAddress.Height + 15

If Len(StartingAddress) > 0 Then

cboAddress.Text = StartingAddress

cboAddress.AddItem cboAddress.Text

'try to navigate to the starting address

timTimer.Enabled = True

brwWebBrowser.Navigate StartingAddress

End If

End Sub

Private Sub brwWebBrowser_DownloadComplete()

On Error Resume Next

Me.Caption = brwWebBrowser.LocationName

End Sub

Private Sub brwWebBrowser_NavigateComplete(ByVal URL As String)

Dim i As Integer


```

Dim bFound As Boolean
Me.Caption = brwWebBrowser.LocationName
For i = 0 To cboAddress.ListCount - 1
    If cboAddress.List(i) = brwWebBrowser.LocationURL Then
        bFound = True
        Exit For
    End If
Next i
mbDontNavigateNow = True
If bFound Then
    cboAddress.RemoveItem i
End If
cboAddress.AddItem brwWebBrowser.LocationURL, 0
cboAddress.ListIndex = 0
mbDontNavigateNow = False
End Sub

Private Sub cboAddress_Click()
    If mbDontNavigateNow Then Exit Sub
    timTimer.Enabled = True
    brwWebBrowser.Navigate cboAddress.Text
End Sub

Private Sub cboAddress_KeyPress(KeyAscii As Integer)
    On Error Resume Next
    If KeyAscii = vbKeyReturn Then
        cboAddress_Click
    End If
End Sub

Private Sub Form_Resize()
    cboAddress.Width = Me.ScaleWidth - 100
    brwWebBrowser.Width = Me.ScaleWidth - 100
    brwWebBrowser.Height = Me.ScaleHeight - (picAddress.Top +
        picAddress.Height) - 100

```

End Sub

Private Sub timTimer_Timer()

 If brwWebBrowser.Busy = False Then

 timTimer.Enabled = False

 Me.Caption = brwWebBrowser.LocationName

 Else

 Me.Caption = "Working..."

 End If

End Sub

Private Sub tbToolBar_ButtonClick(ByVal Button As Button)

 On Error Resume Next

 timTimer.Enabled = True

 Select Case Button.Key

 Case "Back"

 brwWebBrowser.GoBack

 Case "Forward"

 brwWebBrowser.GoForward

 Case "Refresh"

 brwWebBrowser.Refresh

 Case "Home"

 brwWebBrowser.GoHome

 Case "Search"

 brwWebBrowser.GoSearch

 Case "Stop"

 timTimer.Enabled = False

 brwWebBrowser.Stop

 Me.Caption = brwWebBrowser.LocationName

 End Select

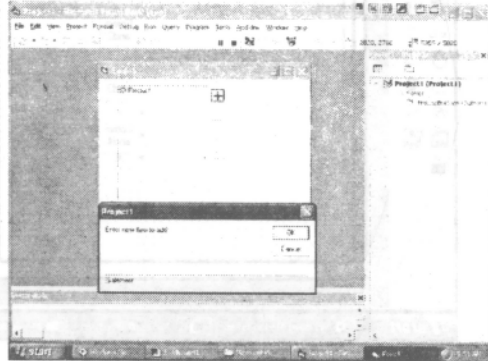
End Sub

Tạo các nút khiển (Control)

Bài 8: Tạo khung danh mục nút lệnh

Trong lần thực hành này, bạn thiết kế một khung danh mục thêm các thành phần cần thiết vào khung danh mục với các nút biểu tượng lệnh Add, Delete, Up và Down.

Trong khung mẫu biểu danh mục này, sau khi kích nút lệnh biểu tượng Add, một khung thoại hiện ra để bạn nhập đối tượng muốn thêm vào khung danh mục, hoặc chọn danh mục trong khung danh sách và kích biểu tượng nút lệnh Delete để xóa, hoặc kích các biểu tượng nút lệnh Up hoặc Down để di chuyển thanh chọn trong khung danh sách.

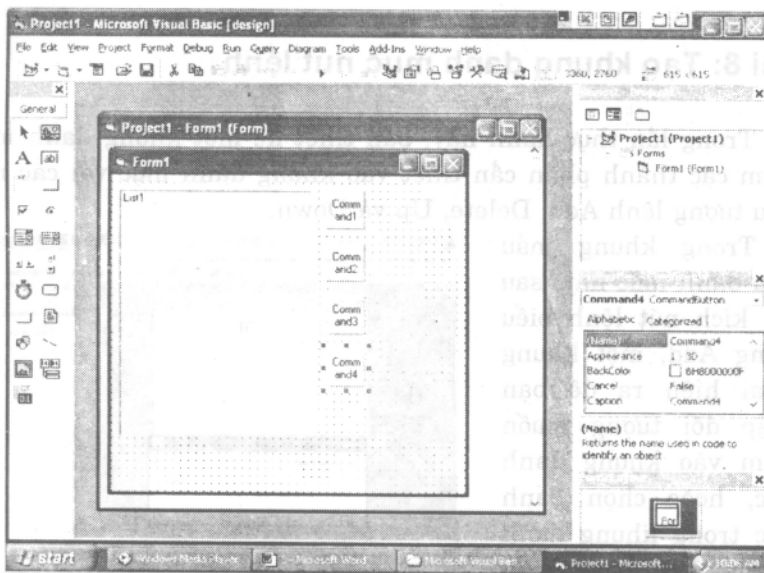


Bạn có thể dựa theo bài tập này để thiết kế thêm hoặc thay đổi mã lệnh trong khung cửa sổ Code theo tác vụ riêng của bạn.

1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích công cụ ListBox trong thanh công cụ General. Kích và kéo mouse trong khung cửa sổ Form để tạo khung có kích thước theo ý muốn.
4. Lần lượt nút lệnh CommandButton, tìm vị trí và tạo bốn nút lệnh có kích thước bằng nhau nằm theo chiều dọc, bên cạnh phải khung ListBox.



Và mẫu biểu thiết kế ban đầu có dạng như sau:



Gán thuộc tính đối tượng

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	frmListButtons
Caption	Form1

Gán thuộc tính khung ListBox

Font	MS Sans Serif
Name	lstItems
ItemData	(List)
List	(List)

Gán thuộc tính nút lệnh Command1

Name	cmdAdd
------	--------

Picture Bitmap
Style 1 – Graphical

Gán thuộc tính nút lệnh Command2

Name cmdDelete
Picture Bitmap
Style 1 – Graphical

Gán thuộc tính nút lệnh Command3

Name cmdUp
Picture Bitmap
Style 1 – Graphical

Gán thuộc tính nút lệnh Command4

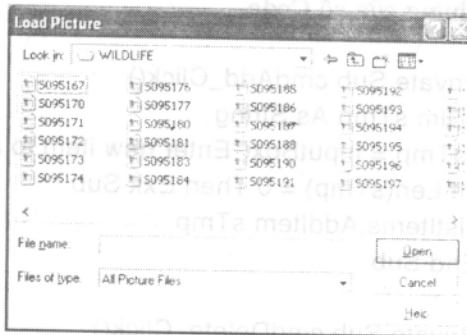
Name cmdDown
Picture Bitmap
Style 1 – Graphical

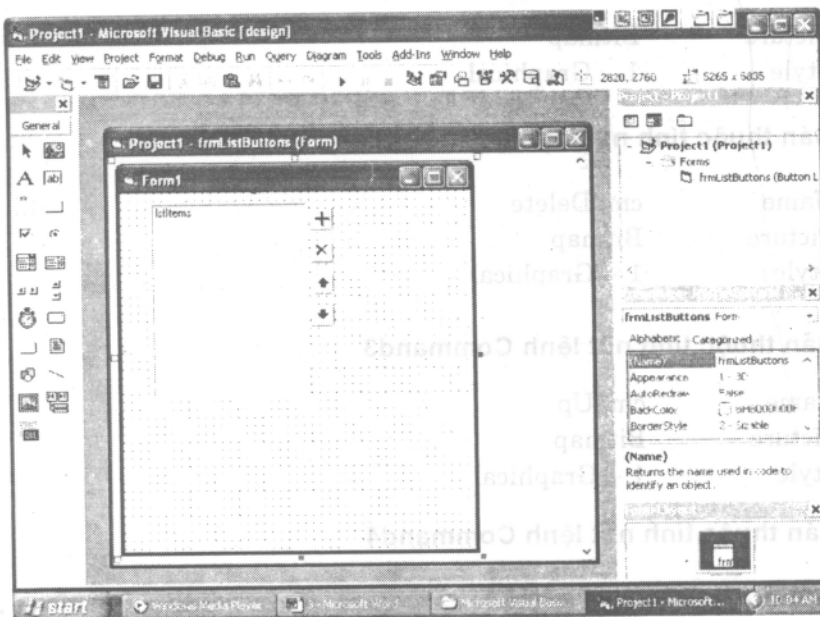
Ghi chú: Muốn tạo nút

lệnh theo biểu tượng
như hình minh họa,
kích chọn nút lệnh.

Trong khung cửa sổ
Properties, tìm thuộc
tính Picture trong cột
bên trái, kích nút
mouse vào cột bên
phải, tìm thư mục
Icon trong khung

thoại Load Picture và chọn tên tập tin biểu tượng. Kích Open
để tạo nút lệnh theo biểu tượng thay vì ký tự.





Sau khi gán thuộc tính, thay đổi tên các đối tượng trong mẫu biểu, kích đúp vào từng đối tượng và lập trình cho chúng trong khung cửa sổ Code.

```
Private Sub cmdAdd_Click()
```

```
    Dim sTmp As String
```

```
    sTmp = InputBox("Enter new item to add:")
```

```
    If Len(sTmp) = 0 Then Exit Sub
```

```
    lstItems.AddItem sTmp
```

```
End Sub
```

```
Private Sub cmdDelete_Click()
```

```
    If lstItems.ListIndex > -1 Then
```

```
        If MsgBox("Delete '" & lstItems.Text & "'", vbQuestion +  
            vbYesNo) = vbYes Then
```

```
            lstItems.RemoveItem lstItems.ListIndex
```

```
        End If
```

```
End If
End Sub
```

```
Private Sub cmdUp_Click()
    On Error Resume Next
    Dim nItem As Integer

    With lstItems
        If .ListIndex < 0 Then Exit Sub
        nItem = .ListIndex
        If nItem = 0 Then Exit Sub 'can't move 1st item up
        'move item up
        .AddItem .Text, nItem - 1
        'remove old item
        .RemoveItem nItem + 1
        'select the item that was just moved
        .Selected(nItem - 1) = True
    End With
End Sub
```

```
Private Sub cmdDown_Click()
    On Error Resume Next
    Dim nItem As Integer

    With lstItems
        If .ListIndex < 0 Then Exit Sub
        nItem = .ListIndex
        If nItem = .ListCount - 1 Then Exit Sub 'can't move last item
down
        'move item down
        .AddItem .Text, nItem + 2
        'remove old item
        .RemoveItem nItem
        'select the item that was just moved
        .Selected(nItem + 1) = True
    End With
End Sub
```

```
End With  
End Sub
```

```
Private Sub lstItems_DragDrop(Source As Control, X As Single, Y  
As Single)
```

```
Dim i As Integer  
Dim nID As Integer  
Dim sTmp As String
```

```
If Source.Name <> "lstItems" Then Exit Sub  
If lstItems.ListCount = 0 Then Exit Sub
```

```
With lstItems
```

```
    i = (Y \ TextHeight("A")) + .TopIndex
```

```
    If i = .ListIndex Then  
        'dropped on top of itself  
        Exit Sub
```

```
    End If
```

```
    If i > .ListCount - 1 Then i = .ListCount - 1
```

```
    nID = .ListIndex
```

```
    sTmp = .Text
```

```
    If (nID > -1) Then
```

```
        sTmp = .Text
```

```
        .RemoveItem nID
```

```
        .AddItem sTmp, i
```

```
        .ListIndex = .NewIndex
```

```
    End If
```

```
End With
```

```
SetListButtons
```

```
End Sub
```

```
Sub lstItems_MouseMove(Button As Integer, Shift As Integer, X  
As Single, Y As Single)
```

```
    If Button = vbLeftButton Then lstItems.Drag
```

```
End Sub
```



```
Private Sub lstItems_Click()
```

```
    SetListButtons
```

```
End Sub
```

```
Sub SetListButtons()
```

```
    Dim i As Integer
```

```
    i = lstItems.ListIndex
```

```
    'set the state of the move buttons
```

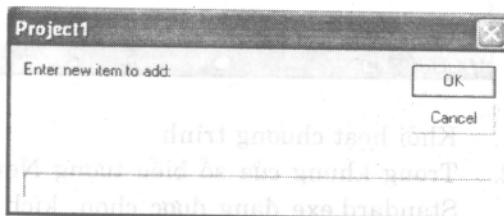
```
    cmdUp.Enabled = (i > 0)
```

```
    cmdDown.Enabled = ((i > -1) And (i < (lstItems.ListCount - 1)))
```

```
    cmdDelete.Enabled = (i > -1)
```

```
End Sub
```

Sau khi gõ rồi và để điều hoạt, ấn phím F5. Trong khung cửa sổ điều hoạt, kích nút lệnh Add và màn hình sẽ hiện khung thoại yêu cầu cho biết



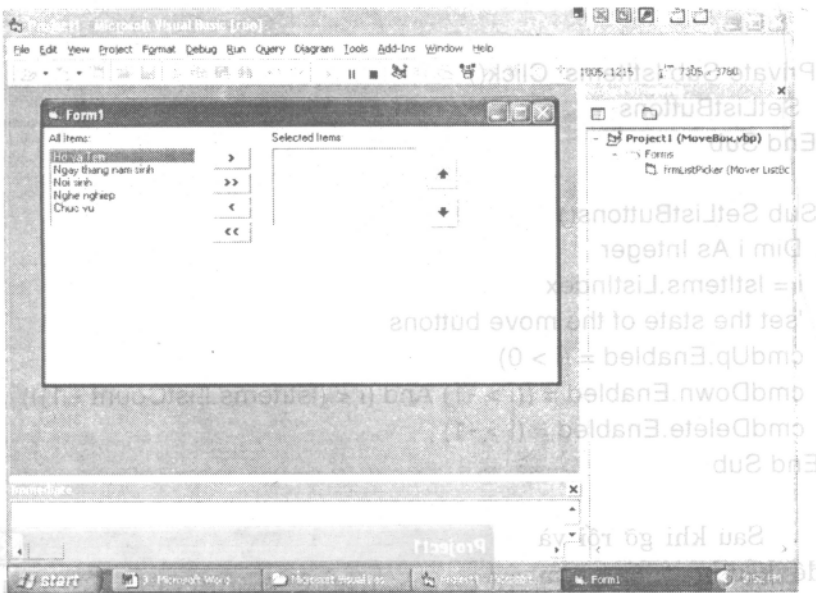
tên đối tượng muốn thêm. Sau khi nhập tên đối tượng và kích OK, tên đối tượng sẽ được thêm vào khung danh sách.

Nếu muốn xóa, kích chọn tên đối tượng trong khung danh sách và kích nút lệnh Delete.

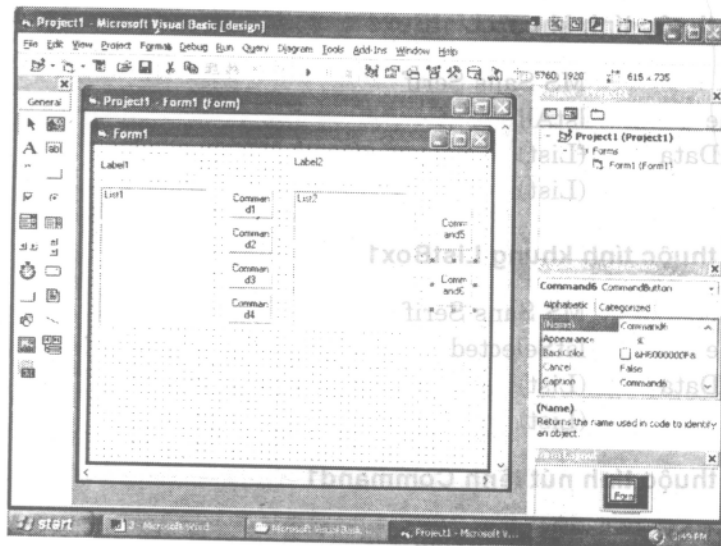
Bạn có thể thay đổi mã chương trình theo nhu cầu tác vụ.

Bài 9: Thiết kế khung chuyển danh sách (MoveList)

Trong bài tập này, bạn thiết kế khung danh sách chọn chuyển đối tượng để tạo một mẫu biểu dữ liệu quản lý nhân sự như từng thường gặp trong các Wizard của những chương trình cơ sở dữ liệu khi muốn lập mẫu biểu.



1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích công cụ Label trong thanh công cụ General. Kích chọn vị trí trong khung cửa sổ thiết kế Form và kéo để tạo khung nhãn Label1.
4. Lặp lại bước 3 để tạo khung nhãn Label2.
5. Kích công cụ ListBox trong thanh công cụ General. Kích và kéo mouse trong khung cửa sổ Form để tạo khung có kích thước theo ý muốn.
6. Lặp lại bước 5 để tạo khung ListBox2.
7. Lần lượt nút lệnh CommandButton, tìm vị trí và tạo bốn nút lệnh có kích thước bằng nhau nằm theo chiều dọc, bên cạnh phải khung ListBox1.
8. Lặp lại bước 7 để tạo hai nút lệnh có kích thước bằng nhau nằm theo chiều dọc, bên cạnh phải khung ListBox2.



Gán thuộc tính đối tượng

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	frmListPicker
Caption	Form1

Gán thuộc tính khung Label1

Font	MS Sans Serif
Name	lblAll
Alignment	Left
Caption	All Items:

Gán thuộc tính khung Label2

Font	MS Sans Serif
Name	lblSelected
Alignment	Left
Caption	Selected Items:

Gán thuộc tính khung ListBox1

Font	MS Sans Serif
Name	lstAll
ItemData	(List)
List	(List)

Gán thuộc tính khung ListBox1

Font	MS Sans Serif
Name	lstSelected
ItemData	(List)
List	(List)

Gán thuộc tính nút lệnh Command1

Name	cmdRightOne
Caption	> (ký hiệu > trên bàn phím)

Gán thuộc tính nút lệnh Command2

Name	cmdRightAll
Caption	>>

Gán thuộc tính nút lệnh Command3

Name	cmdLeft
Caption	< (ký hiệu < trên bàn phím)

Gán thuộc tính nút lệnh Command4

Name	cmdAllLeft
Caption	<< (ký hiệu < trên bàn phím)

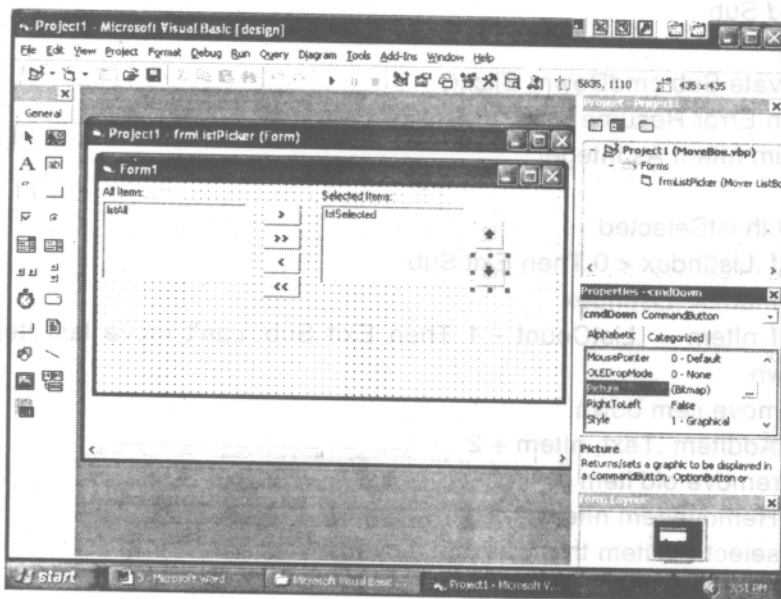
Gán thuộc tính nút lệnh Command5

Name	cmdUp
------	-------

Picture Bitmap
Style 1 – Graphical

Gán thuộc tính nút lệnh Command6

Name cmdDown
Picture Bitmap
Style 1 – Graphical



Kích đúp vào từng đối tượng và lập trình cho chúng trong khung cửa sổ Code:

```
Private Sub cmdUp_Click()  
On Error Resume Next  
Dim nItem As Integer  
With lstSelected  
If .ListIndex < 0 Then Exit Sub  
nItem = .ListIndex
```

```

    If nItem = 0 Then Exit Sub 'can't move 1st item up
    'move item up
    .AddItem .Text, nItem - 1
    'remove old item
    .RemoveItem nItem + 1
    'select the item that was just moved
    .Selected(nItem - 1) = True
End With
End Sub

Private Sub cmdDown_Click()
    On Error Resume Next
    Dim nItem As Integer

    With lstSelected
        If .ListIndex < 0 Then Exit Sub
        nItem = .ListIndex
        If nItem = .ListCount - 1 Then Exit Sub 'can't move last item
        down
        'move item down
        .AddItem .Text, nItem + 2
        'remove old item
        .RemoveItem nItem
        'select the item that was just moved
        .Selected(nItem + 1) = True
    End With
End Sub

Private Sub cmdRightOne_Click()
    On Error Resume Next
    Dim i As Integer

    If lstAll.ListCount = 0 Then Exit Sub

    lstSelected.AddItem lstAll.Text

```

```

i = lstAll.ListIndex
lstAll.RemoveItem lstAll.ListIndex
If lstAll.ListCount > 0 Then
    If i > lstAll.ListCount - 1 Then
        lstAll.ListIndex = i - 1
    Else
        lstAll.ListIndex = i
    End If
End If
lstSelected.ListIndex = lstSelected.NewIndex
End Sub

```

```

Private Sub cmdRightAll_Click()
    On Error Resume Next
    Dim i As Integer
    For i = 0 To lstAll.ListCount - 1
        lstSelected.AddItem lstAll.List(i)
    Next
    lstAll.Clear
    lstSelected.ListIndex = 0
End Sub

```

```

Private Sub cmdLeftOne_Click()
    On Error Resume Next
    Dim i As Integer

    If lstSelected.ListCount = 0 Then Exit Sub

    lstAll.AddItem lstSelected.Text
    i = lstSelected.ListIndex
    lstSelected.RemoveItem i

    lstAll.ListIndex = lstAll.NewIndex
    If lstSelected.ListCount > 0 Then
        If i > lstSelected.ListCount - 1 Then

```

```

        lstSelected.ListIndex = i - 1
    Else
        lstSelected.ListIndex = i
    End If
End If
End Sub
Private Sub cmdLeftAll_Click()
    On Error Resume Next
    Dim i As Integer
    For i = 0 To lstSelected.ListCount - 1
        lstAll.AddItem lstSelected.List(i)
    Next
    lstSelected.Clear
    lstAll.ListIndex = lstAll.NewIndex

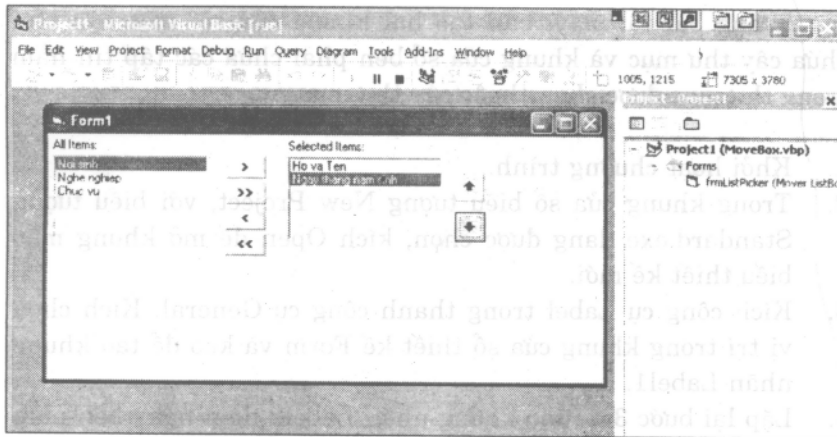
End Sub

Private Sub Form_Load()
    lstAll.AddItem "Ho va Ten"
    lstAll.AddItem "Ngay thang nam sinh"
    lstAll.AddItem "Noi sinh"
    lstAll.AddItem "Nghe nghiep"
    lstAll.AddItem "Chuc vu"
    lstAll.ListIndex = 0
End Sub

Private Sub lstAll_DblClick()
    cmdRightOne_Click
End Sub

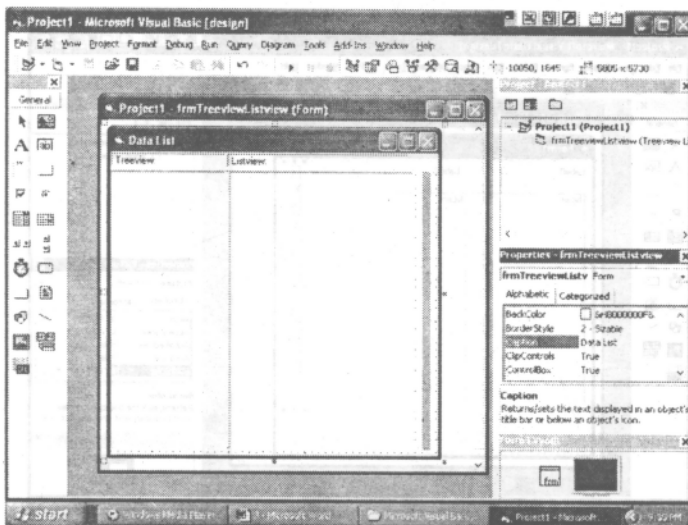
Private Sub lstSelected_DblClick()
    cmdLeftOne_Click
End Sub

```

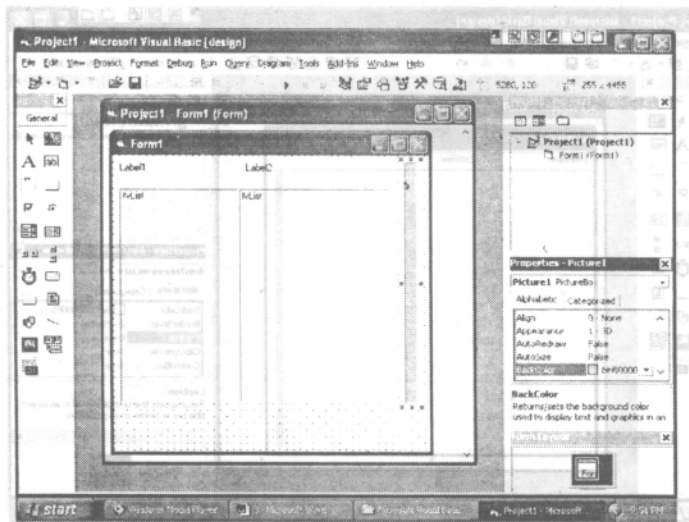
Sau khi kết thúc và để kiểm tra, ấn phím F5. Trong khung màn hình điều hoạt, kích chọn thành phần trong khung cửa sổ All Items và kích nút lệnh chuyển đơn. Thành phần vừa chọn sẽ chuyển sang khung cửa sổ Selected Items.

Bài 10: Tạo khung cửa sổ cây danh mục dữ liệu



Trong bài tập này, bạn tạo hai khung cửa sổ: một bên trái chứa cây thư mục và khung cửa sổ bên phải chứa các tập tin nằm trong thư mục được chọn thuộc cây thư mục.

1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích công cụ Label trong thanh công cụ General. Kích chọn vị trí trong khung cửa sổ thiết kế Form và kéo để tạo khung nhãn Label1.
4. Lặp lại bước 3 để tạo khung nhãn Label2 nằm ngay bên phải khung nhãn Label1.
5. Kích công cụ TreeViewBox trong thanh công cụ General. Kích và kéo mouse trong khung cửa sổ Form để tạo khung có kích thước theo ý muốn.
6. Kích công cụ ListViewBox trong thanh công cụ General. Kích và kéo mouse trong khung cửa sổ Form để tạo khung có kích thước theo ý muốn ngay cạnh phải của khung cây thư mục.



7. Kích công cụ PictureBox trong thanh công cụ General. Kích và kéo mouse trong khung cửa sổ Form để tạo khung ảnh có kích thước theo ý muốn ngay cạnh phải của khung danh sách tập tin.

Gán thuộc tính đối tượng

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	frmTreeViewListView
Caption	Data List

Gán thuộc tính khung Label1

Font	MS Sans Serif
Name	lblTitle
Alignment	Left
Caption	Treeview:

Gán thuộc tính khung Label2

Font	MS Sans Serif
Name	lblTitle
Alignment	Left
Caption	Listview:

Gán thuộc tính khung TreeViewBox

Font	MS Sans Serif
Name	tvTreeView
BackColor	xám

Gán thuộc tính khung ListViewBox

Font	MS Sans Serif
Name	lvListView
BackColor	trắng

Gán thuộc tính nút lệnh PictureBox

Name	picSplitter
FillColor	xám đậm

Sau khi khai báo và thay đổi các nhãn và thuộc tính đối tượng, kích đúp vào từng đối tượng và lập trình cho chúng như sau:

```
Const NAME_COLUMN = 0
Const TYPE_COLUMN = 1
Const SIZE_COLUMN = 2
Const DATE_COLUMN = 3
```

```
Dim mbMoving As Boolean
Const sgfSplitLimit = 500
```

```
Private Sub Form_Resize()
    If Me.Width < 3000 Then Me.Width = 3000
    SizeControls imgSplitter.Left
End Sub
```

```
Private Sub imgSplitter_MouseDown(Button As Integer, Shift As
                                   Integer, X As Single, Y As
                                   Single)
    With imgSplitter
        picSplitter.Move .Left, .Top, .Width - 20, .Height - 20
    End With
    picSplitter.Visible = True
    mbMoving = True
End Sub
```

```
Private Sub imgSplitter_MouseMove(Button As Integer, Shift As
                                   Integer, X As Single, Y As
                                   Single)
```

Dim sglPos As Single

```

If mbMoving Then
    sglPos = X + imgSplitter.Left
    If sglPos < sglSplitLimit Then
        picSplitter.Left = sglSplitLimit
    ElseIf sglPos > Me.Width - sglSplitLimit Then
        picSplitter.Left = Me.Width - sglSplitLimit
    Else
        picSplitter.Left = sglPos
    End If
End If
End Sub

```

```
Private Sub imgSplitter_MouseUp(Button As Integer, Shift As
                                Integer, X As Single, Y As Single)
    SizeControls picSplitter.Left
    picSplitter.Visible = False
    mbMoving = False
End Sub
```

```
Sub SizeControls(X As Single)
    On Error Resume Next

    'set the width
    If X < 1500 Then X = 1500
    If X > (Me.Width - 1500) Then X = Me.Width - 1500
    tvTreeView.Width = X
    imgSplitter.Left = X
    lvListView.Left = X + 40
    lvListView.Width = Me.Width - (tvTreeView.Width + 140)
    lblTitle(0).Width = tvTreeView.Width
    lblTitle(1).Left = lvListView.Left + 20
    lblTitle(1).Width = lvListView.Width - 40
End Sub
```

```

'set the top
' If tbToolBar.Visible Then
'   tvTreeView.Top = tbToolBar.Height + picTitles.Height
' Else
'   tvTreeView.Top = picTitles.Height
' End If
lvListView.Top = tvTreeView.Top

'set the height
' If sbStatusBar.Visible Then
'   tvTreeView.Height = Me.ScaleHeight - (picTitles.Top +
'                                       picTitles.Height +
'                                       sbStatusBar.Height)
' Else
'   tvTreeView.Height = Me.ScaleHeight - (picTitles.Top +
'                                       picTitles.Height) ' + h)
' End If

lvListView.Height = tvTreeView.Height
imgSplitter.Top = tvTreeView.Top
imgSplitter.Height = tvTreeView.Height
End Sub

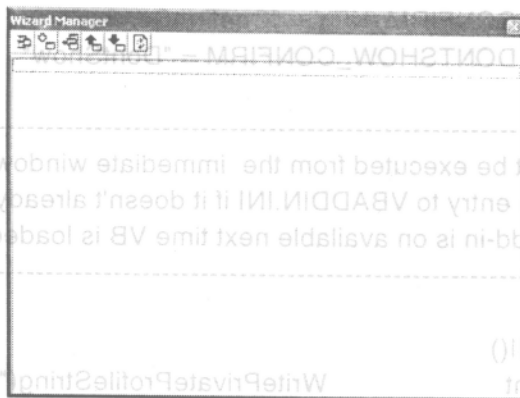
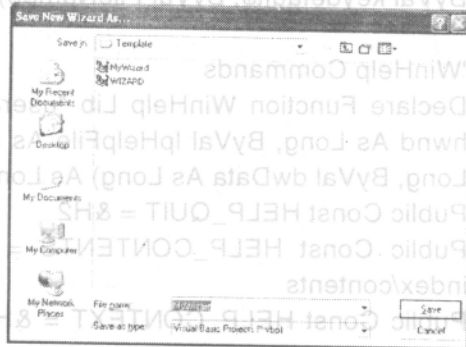
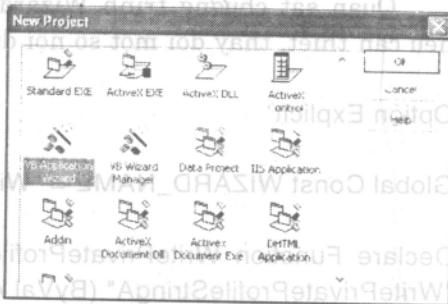
Private Sub tvTreeView_DragDrop(Source As Control, X As
                               Single, Y As Single)
    If Source = imgSplitter Then
        SizeControls X
    End If
End Sub

```

Tạo khung Wizard

Trước khi tạo các khung hướng dẫn Wizard, bạn phải tạo một thư mục quản lý riêng bằng cách:

1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, kích đúp biểu tượng VB Application Wizard để mở khung thiết kế mới các đối tượng hướng dẫn.
3. Kích màn hình hiện khung thoại thông báo là không tìm thấy đề án của Wizard và hỏi bạn có tạo đối tượng quản lý Wizard mới hay không, kích Yes để chấp nhận và màn hình hiện khung thoại Save As để bạn lưu lại theo tên mặc định hoặc tên tự chọn.
4. Sau khi kích Yes, màn hình hiện khung cửa sổ quản lý các mẫu biểu hướng dẫn.



Quan sát chương trình Wizard trong khung của sổ Code và nếu cần thiết, thay đổi một số nội dung theo nhu cầu tác vụ.

Option Explicit

Global Const WIZARD_NAME = "WizardTemplate"

Declare Function WritePrivateProfileString& Lib "Kernel32" Alias
"WritePrivateProfileStringA" (ByVal AppName\$, ByVal KeyName\$,
ByVal keydefault\$, ByVal FileName\$)

'WinHelp Commands

Declare Function WinHelp Lib "user32" Alias "WinHelpA" (ByVal
hwnd As Long, ByVal lpHelpFile As String, ByVal wCommand As
Long, ByVal dwData As Long) As Long

Public Const HELP_QUIT = &H2 ' Terminate help

Public Const HELP_CONTENTS = &H3& ' Display
index/contents

Public Const HELP_CONTEXT = &H1 ' Display topic in
ulTopic

Public Const HELP_INDEX = &H3 ' Display index

Global Const APP_CATEGORY = "Wizards"

Global Const CONFIRM_KEY = "ConfirmScreen"

Global Const DONTSHOW_CONFIRM = "DontShow"

'this sub must be executed from the immediate window
'it will add the entry to VBADDIN.INI if it doesn't already exist
'so that the add-in is on available next time VB is loaded

Sub AddToINI()

 Debug.Print WritePrivateProfileString("Add-Ins32",


```
WIZARD_NAME & ".Wizard", "0", "VBADDIN.INI")  
End Sub
```

```
Function GetResString(nRes As Integer) As String  
    Dim sTmp As String  
    Dim sRetStr As String  
  
    Do  
        sTmp = LoadResString(nRes)  
        If Right(sTmp, 1) = "_" Then  
            sRetStr = sRetStr + VBA.Left(sTmp, Len(sTmp) - 1)  
        Else  
            sRetStr = sRetStr + sTmp  
        End If  
        nRes = nRes + 1  
    Loop Until Right(sTmp, 1) <> "_"  
    GetResString = sRetStr  
  
End Function
```

```
Function GetField(sBuffer As String, sSep As String) As String  
    Dim p As Integer  
  
    p = InStr(sBuffer & sSep, sSep)  
    GetField = VBA.Left(sBuffer, p - 1)  
    sBuffer = Mid(sBuffer, p + Len(sSep))  
  
End Function
```

```
Sub LoadResStrings(frm As Form)  
    On Error Resume Next  
  
    Dim ctl As Control  
    Dim obj As Object
```

```

'set the form's caption
If IsNumeric(frm.Tag) Then
    frm.Caption = LoadResString(CInt(frm.Tag))
End If

'set the controls' captions using the caption
'property for menu items and the Tag property
'for all other controls
For Each ctl In frm.Controls
    If TypeName(ctl) = "Menu" Then
        If IsNumeric(ctl.Caption) Then
            If Err = 0 Then
                ctl.Caption = LoadResString(CInt(ctl.Caption))
            Else
                Err = 0
            End If
        End If
    ElseIf TypeName(ctl) = "TabStrip" Then
        For Each obj In ctl.Tabs
            If IsNumeric(obj.Tag) Then
                obj.Caption = LoadResString(CInt(obj.Tag))
            End If
            'check for a tooltip
            If IsNumeric(obj.ToolTipText) Then
                If Err = 0 Then
                    obj.ToolTipText = LoadResString(CInt(obj.ToolTipText))
                Else
                    Err = 0
                End If
            End If
        Next
    ElseIf TypeName(ctl) = "Toolbar" Then
        For Each obj In ctl.Buttons
            If IsNumeric(obj.Tag) Then
                obj.ToolTipText = LoadResString(CInt(obj.Tag))
            End If
        Next
    End If
Next

```

```

        End If
    Next
ElseIf TypeName(ctl) = "ListView" Then
    For Each obj In ctl.ColumnHeaders
        If IsNumeric(obj.Tag) Then
            obj.Text = LoadResString(CInt(obj.Tag))
        End If
    Next
Else
    If IsNumeric(ctl.Tag) Then
        If Err = 0 Then
            ctl.Caption = GetResString(CInt(ctl.Tag))
        Else
            Err = 0
        End If
    End If
    'check for a tooltip
    If IsNumeric(ctl.ToolTipText) Then
        If Err = 0 Then
            ctl.ToolTipText = LoadResString(CInt(ctl.ToolTipText))
        Else
            Err = 0
        End If
    End If
End If
Next

End Sub

'=====
'Purpose: Replace the <TOPIC_TEXT> string(s) in
'         res file string for correct placement
'         of localized tokens
'
'Inputs: sString = String to search and replace in

```

' sReplacement = String to replace token with
' sReplacement2 = 2nd String to replace token with
'

'Outputs: New string with token replaced throughout

'=====

```
Function ReplaceTopicTokens(sString As String, _  
    sReplacement As String, _  
    sReplacement2 As String) As String
```

```
On Error Resume Next
```

```
Dim p As Integer  
Dim sTmp As String
```

```
Const TOPIC_TEXT = "<TOPIC_TEXT>"  
Const TOPIC_TEXT2 = "<TOPIC_TEXT2>"
```

```
sTmp = sString
```

```
Do
```

```
    p = InStr(sTmp, TOPIC_TEXT)
```

```
    If p Then
```

```
        sTmp = VBA.Left(sTmp, p - 1) + sReplacement +  
                Mid(sTmp, p + Len(TOPIC_TEXT))
```

```
    End If
```

```
Loop While p
```

```
If Len(sReplacement2) > 0 Then
```

```
    Do
```

```
        p = InStr(sTmp, TOPIC_TEXT2)
```

```
        If p Then
```

```
            sTmp = VBA.Left(sTmp, p - 1) + sReplacement2 +  
                    Mid(sTmp, p + Len(TOPIC_TEXT2))
```

```
        End If
```

```
    Loop While p
```

```
End If
```

ReplaceTopicTokens = sTmp

End Function

Public Function GetResData(sResName As String, sResType As String) As String

Dim sTemp As String

Dim p As Integer

sTemp = StrConv(LoadResData(sResName, sResType), vbUnicode)

p = InStr(sTemp, vbNullChar)

If p Then sTemp = VBA.Left\$(sTemp, p - 1)

GetResData = sTemp

End Function

Function AddToAddInCommandBar(VBInst As Object, sCaption As String, oBitmap As Object) As Object
'Office.CommandBarControl

On Error GoTo AddToAddInCommandBarErr

Dim c As Integer

Dim cbMenuCommandBar As Object

'Office.CommandBarControl 'command bar object

Dim cbMenu As Object

'see if we can find the Add-Ins menu

Set cbMenu = VBInst.CommandBars("Add-Ins")

If cbMenu Is Nothing Then

'not available so we fail

Exit Function

End If

'add it to the command bar

Set cbMenuCommandBar = cbMenu.Controls.Add(1)

```

c = cbMenu.Controls.Count - 1
If cbMenu.Controls(c).BeginGroup And _
    Not cbMenu.Controls(c - 1).BeginGroup Then
    'this s the first addin being added so it needs a separator
    cbMenuCommandBar.BeginGroup = True
End If

```

```

'set the caption
cbMenuCommandBar.Caption = sCaption
'undone;set the onaction (required at this point)
cbMenuCommandBar.OnAction = "hello"
'copy the icon to the clipboard
Clipboard.SetData oBitmap
'set the icon for the button
cbMenuCommandBar.PasteFace

```

```

Set AddToAddInCommandBar = cbMenuCommandBar

```

```

Exit Function

```

```

AddToAddInCommandBarErr:

```

```

End Function

```

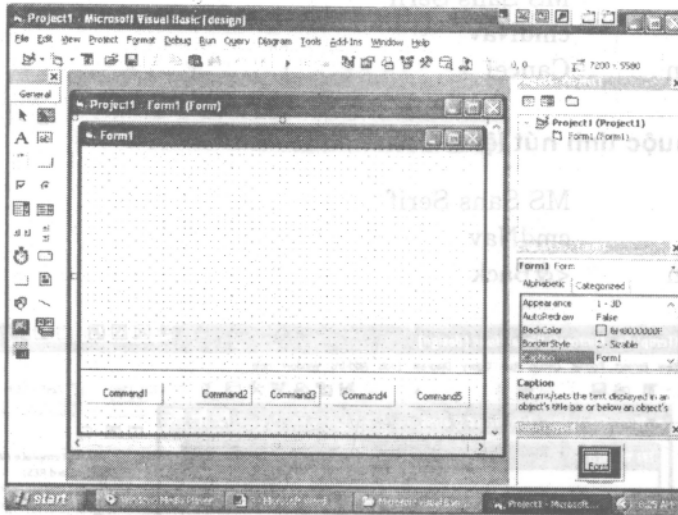
Bài 11: Tạo khung thoại Wizard mẫu

Trong bài tập này, với một khung mẫu Wizard trống đã được lập trình sẵn, bạn có thể đổi tên và tùy thêm các nội dung bên trong mẫu biểu và sau đó lập trình thêm cho chúng trong khung cửa sổ Code.

1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích công cụ Line trong thanh công cụ, chọn vị trí gần cuối

khung mẫu biểu Form, kích chìm nút mouse và kéo từ trái sang phải để tạo đường phân cách.

4. Kích công cụ CommandButton trong thanh công cụ General. Kích và kéo mouse trong khung cửa sổ Form để tạo khung nút lệnh Command1 ngay cuối mẫu biểu.
5. Lặp lại bước 4 và lần lượt tạo thêm bốn nút lệnh nằm thẳng hàng nhau.



Gán thuộc tính đối tượng

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	frmWizard
Caption	??? Wizard

Gán thuộc tính đường kẻ Line

Name	line1
BorderColor	màu xám

Gán thuộc tính nút lệnh Command1

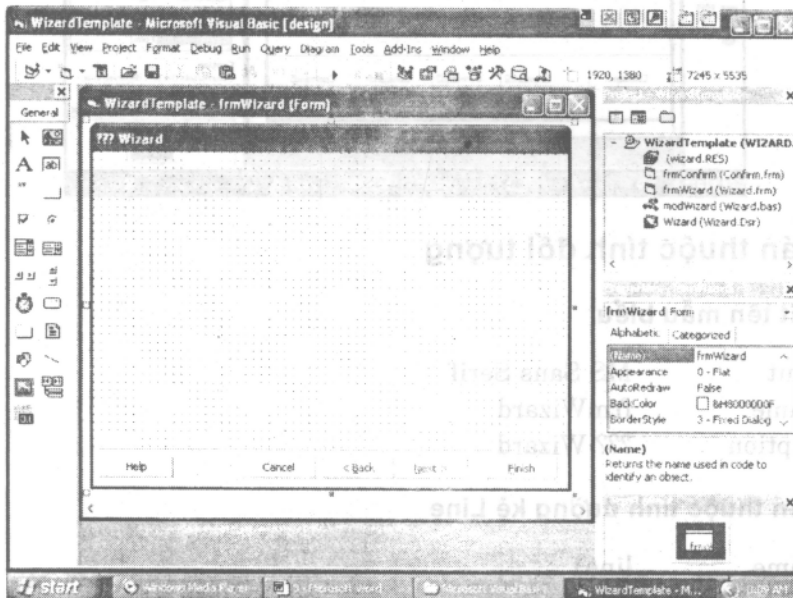
Font	MS Sans Serif
Name	cmdNav
Caption	Help

Gán thuộc tính nút lệnh Command2

Font	MS Sans Serif
Name	cmdNav
Caption	Cancel

Gán thuộc tính nút lệnh Command3

Font	MS Sans Serif
Name	cmdNav
Caption	<&Back



Gán thuộc tính nút lệnh Command4

Font	MS Sans Serif
Name	cmdNav
Caption	&Next>

Gán thuộc tính nút lệnh Command5

Font	MS Sans Serif
Name	cmdNav
Caption	&Finish

Lập mã chương trình trong khung cửa sổ Code.

Option Explicit

```
Const NUM_STEPS = 6
```

```
Const RES_ERROR_MSG = 30000
```

```
'BASE VALUE FOR HELP FILE FOR THIS WIZARD:
```

```
Const HELP_BASE = 1000
```

```
Const HELP_FILE = "MYWIZARD.HLP"
```

```
Const BTN_HELP = 0
```

```
Const BTN_CANCEL = 1
```

```
Const BTN_BACK = 2
```

```
Const BTN_NEXT = 3
```

```
Const BTN_FINISH = 4
```

```
Const STEP_INTRO = 0
```

```
Const STEP_1 = 1
```

```
Const STEP_2 = 2
```

```
Const STEP_3 = 3
```

```
Const STEP_4 = 4
```

```
Const STEP_FINISH = 5
```

Const DIR_NONE = 0

Const DIR_BACK = 1

Const DIR_NEXT = 2

Const FRM_TITLE = "Blank Wizard"

Const INTRO_KEY = "IntroductionScreen"

Const SHOW_INTRO = "ShowIntro"

Const TOPIC_TEXT = "<TOPIC_TEXT>"

'module level vars

Dim mnCurStep As Integer

Dim mbHelpStarted As Boolean

Public VBInst As VBIDE.VBE

Dim mbFinishOK As Boolean

Private Sub chkShowIntro_Click()

 If chkShowIntro.Value Then

 SaveSetting APP_CATEGORY, WIZARD_NAME,
INTRO_KEY, SHOW_INTRO

 Else

 SaveSetting APP_CATEGORY, WIZARD_NAME,
INTRO_KEY, vbNullString

 End If

End Sub

Private Sub cmdNav_Click(Index As Integer)

 Dim nAltStep As Integer

 Dim iHelpTopic As Long

 Dim rc As Long

 Select Case Index

 Case BTN_HELP

 mbHelpStarted = True

 iHelpTopic = HELP_BASE + 10 * (1 + mnCurStep)

```

        rc = WinHelp(Me.hwnd, HELP_FILE, HELP_CONTEXT,
                    IHelpTopic)

Case BTN_CANCEL
    Unload Me

Case BTN_BACK
    'place special cases here to jump
    'to alternate steps
    nAltStep = mnCurStep - 1
    SetStep nAltStep, DIR_BACK

Case BTN_NEXT
    'place special cases here to jump
    'to alternate steps
    nAltStep = mnCurStep + 1
    SetStep nAltStep, DIR_NEXT

Case BTN_FINISH
    'wizard creation code goes here
    Unload Me

If      GetSetting(APP_CATEGORY,      WIZARD_NAME,
        CONFIRM_KEY, vbNullString) = vbNullString Then
    frmConfirm.Show vbModal
End If

End Select
End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As
Integer)
    If KeyCode = vbKeyF1 Then
        cmdNav_Click BTN_HELP
    End If

```

End Sub

Private Sub Form_Load()

Dim i As Integer

'init all vars

mbFinishOK = False

For i = 0 To NUM_STEPS - 1

fraStep(i).Left = -10000

Next

'Load All string info for Form

LoadResStrings Me

'Determine 1st Step:

If GetSetting(APP_CATEGORY, WIZARD_NAME,
INTRO_KEY, vbNullString) = SHOW_INTRO Then

chkShowIntro.Value = vbChecked

SetStep 1, DIR_NEXT

Else

SetStep 0, DIR_NONE

End If

End Sub

Private Sub SetStep(nStep As Integer, nDirection As Integer)

Select Case nStep

Case STEP_INTRO

Case STEP_1

Case STEP_2

Case STEP_3

Case STEP_4

mbFinishOK = False

Case STEP_FINISH

mbFinishOK = True

End Select

'move to new step

fraStep(mnCurStep).Enabled = False

fraStep(nStep).Left = 0

If nStep <> mnCurStep Then

fraStep(mnCurStep).Left = -10000

End If

fraStep(nStep).Enabled = True

SetCaption nStep

SetNavBtns nStep

End Sub

Private Sub SetNavBtns(nStep As Integer)

mnCurStep = nStep

If mnCurStep = 0 Then

cmdNav(BTN_BACK).Enabled = False

cmdNav(BTN_NEXT).Enabled = True

ElseIf mnCurStep = NUM_STEPS - 1 Then

cmdNav(BTN_NEXT).Enabled = False

cmdNav(BTN_BACK).Enabled = True

Else

cmdNav(BTN_BACK).Enabled = True

cmdNav(BTN_NEXT).Enabled = True

End If

If mbFinishOK Then

```

        cmdNav(BTN_FINISH).Enabled = True
    Else
        cmdNav(BTN_FINISH).Enabled = False
    End If
End Sub

Private Sub SetCaption(nStep As Integer)
    On Error Resume Next

    Me.Caption = FRM_TITLE & " - " &
LoadResString(fraStep(nStep).Tag)

End Sub

```

```

'=====
'this sub displays an error message when the user has
'not entered enough data to continue
'=====

Sub IncompleteData(nIndex As Integer)
    On Error Resume Next
    Dim sTmp As String

    'get the base error message
    sTmp = LoadResString(RES_ERROR_MSG)
    'get the specific message
    sTmp = sTmp & vbCrLf & LoadResString(RES_ERROR_MSG +
nIndex)
    Beep
    MsgBox sTmp, vbInformation
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
    On Error Resume Next
    Dim rc As Long

```

'see if we need to save the settings

If chkSaveSettings.Value = vbChecked Then

'SaveSetting APP_CATEGORY, WIZARD_NAME, "OptionName",
Option Value

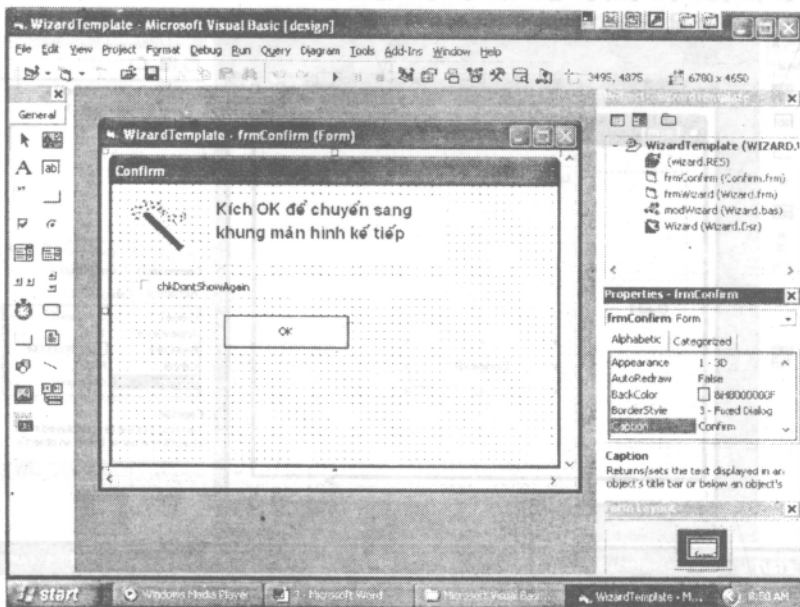
End If

If mbHelpStarted Then rc = WinHelp(Me.hwnd, HELP_FILE,
HELP_QUIT, 0)

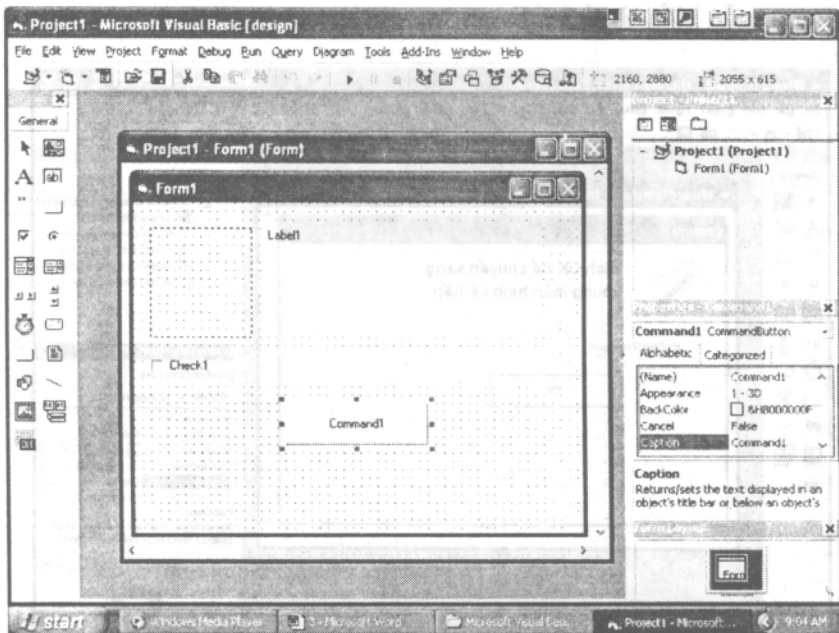
End Sub

Bài 12: Tạo khung thông báo Confirm

Trong bài tập trước, bạn đã làm quen với một khung Wizard trống, và với bài tập này, bạn thêm một số thành phần vào khung mẫu biểu thiết kế Form.



1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích công cụ ImageBox trong thanh công cụ, chọn vị trí gần cuối khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung ảnh.
4. Kích công cụ Label trong thanh công cụ, chọn vị trí gần cuối khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung nhãn Label.
5. Kích công cụ CheckBox trong thanh công cụ, chọn vị trí gần cuối khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung đánh dấu chọn.
6. Kích công cụ CommandButton trong thanh công cụ General. Kích và kéo mouse trong khung cửa sổ Form để tạo khung nút lệnh Command ngay cuối mẫu biểu.



Gán thuộc tính đối tượng

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	frmConfirm
Caption	Confirm

Gán thuộc tính khung ảnh Image

Name	Image1
Picture	Picture

Gán thuộc tính khung Label

Font	MS Sans Serif
Name	lblConfirm
Caption	<i>'Tùy nhập nội dung theo ý muốn'</i>

Gán thuộc tính khung CheckBox

Font	MS Sans Serif
Name	chkDontShowAgain
Caption	chkDontShowAgain

Gán thuộc tính nút lệnh Command

Font	MS Sans Serif
Name	cmdOK
Caption	OK

Chương trình mẫu biểu Confirm trong khung cửa sổ Code:

Option Explicit

```
Private Sub chkDontShowAgain_Click()  
    On Error Resume Next
```

```

If chkDontShowAgain.Value = vbChecked Then
    SaveSetting APP_CATEGORY, WIZARD_NAME,
        CONFIRM_KEY, DONTSHOW_CONFIRM
Else
    SaveSetting APP_CATEGORY, WIZARD_NAME,
        CONFIRM_KEY, vbNullString
End If

End Sub

End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    LoadResStrings Me
End Sub

```

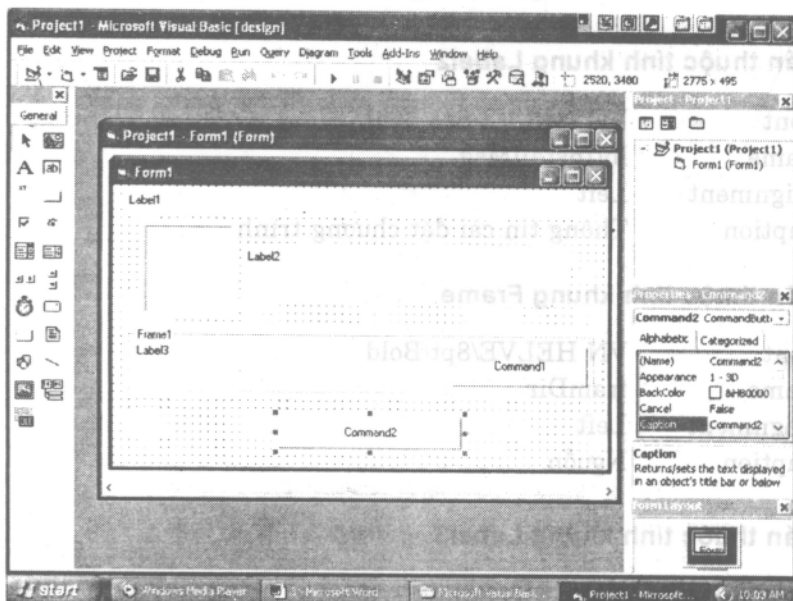
Bài 13: Lập trình mẫu biểu cài đặt chương trình

Đây là bài thực hành mẫu tạo khung thoại cài đặt chương trình đối tượng và bạn có thể dựa theo để thay đổi các tham số thuộc tính cũng như mã lệnh chương trình trong khung cửa sổ Code để thiết kế mẫu biểu theo ý riêng của bạn.

1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích công cụ Label trong thanh công cụ, chọn vị trí gần ngay trên cùng khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung nhãn Label1.
4. Kích công cụ PictureBox trong thanh công cụ, chọn vị trí trong khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung ảnh.

5. Kích công cụ Label trong thanh công cụ, chọn vị trí gần ngay bên phải khung ảnh Picture vừa tạo, kích chìm nút mouse và kéo để tạo khung nhãn Label2.
6. Kích công cụ FrameBox trong thanh công cụ, chọn vị trí gần cuối khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung.
7. Kích công cụ Label trong thanh công cụ, kích chìm nút mouse trong khung FramBox vừa tạo và kéo để tạo khung nhãn Label3.
8. Kích công cụ CommandButton trong thanh công cụ General. Kích và kéo mouse trong khung FrameBox bên cạnh phải khung nhãn Label3 để tạo khung nút lệnh Command1.
9. Kích lại công cụ CommandButton trong thanh công cụ General. Kích và kéo mouse khoảng giữa gần cuối khung mẫu biểu Form để tạo khung nút lệnh Command2.

Mẫu biểu thiết kế ban đầu có dạng như sau:



Gán thuộc tính đối tượng

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	frmBegin
Caption	INSTALLATION

Gán thuộc tính khung Label1

Font	VNI Helve/10pt/Bold
Name	lblBegin
Alignment	Left
Caption	Cài đặt chương trình

Gán thuộc tính khung ảnh Picture

Name	picInstall
Picture	(Bitmap) <i>chọn tập tin hình ảnh từ khung thoại</i>

Gán thuộc tính khung Label2

Font	VN HELVE/8pt/Bold
Name	lblInstallMsg
Alignment	Left
Caption	Thông tin cài đặt chương trình

Gán thuộc tính khung Frame

Font	VN HELVE/8pt/Bold
Name	framDir
Alignment	Left
Caption	Nguồn

Gán thuộc tính khung Label3

Font	VN HELVE/8pt/
------	---------------

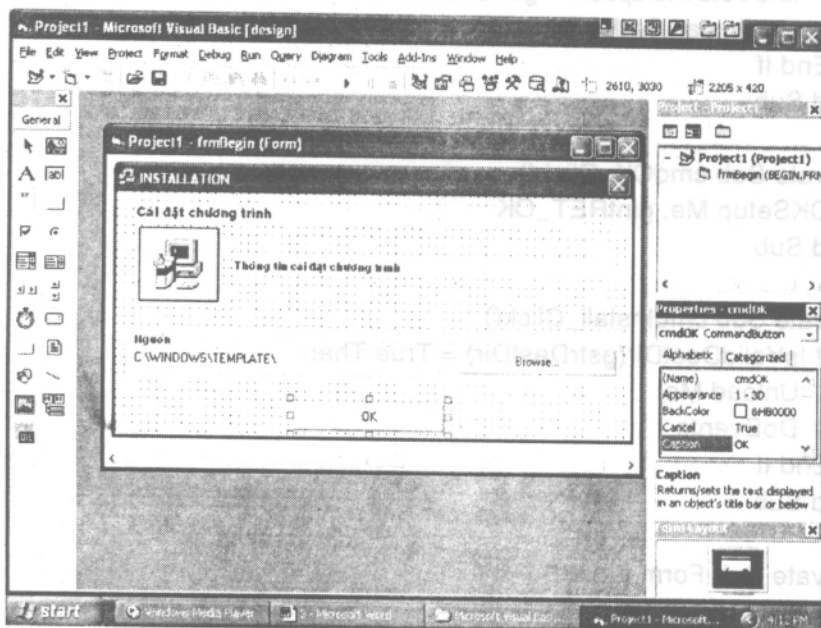
Name	lblDestDir
Alignment	Left
Caption	C:\Windows\Template\

Gán thuộc tính nút lệnh Command1

Font	MS Sans Serif
Name	cmdChDir
Caption	Browser...

Gán thuộc tính nút lệnh Command2

Font	MS Sans Serif
Name	cmdOK
Caption	OK



Chương trình của mẫu biểu trong khung của số Code:

Option Explicit
Option Compare Text

' Constants/Types for a simulated "picture button"
Const x3DLineWidth = 1 "Width" of highlight/shadow "lines"
Const y3DLineWidth = x3DLineWidth
Const pbXDeltaMouseDown = x3DLineWidth
Const pbYDeltaMouseDown = y3DLineWidth

Private fMouseDown As Boolean

Private Sub cmdChDir_Click()
 ShowPathDialog gstrDIR_DEST

 If gfRetVal = gintRET_CONT Then
 lblDestDir.Caption = gstrDestDir
 cmdInstall.SetFocus
 End If
End Sub

Private Sub cmdOK_Click()
 OKSetup Me, gintRET_OK
End Sub

Private Sub cmdInstall_Click()
 If IsValidDestDir(gstrDestDir) = True Then
 Unload Me
 DoEvents
 End If
End Sub

Private Sub Form_Load()
 SetFormFont Me
 fraDir.Caption = ResolveResString(resFRMDIRECTORY)
 cmdChDir.Caption = ResolveResString(resBTNCHGDIR)

```

cmdExit.Caption = ResolveResString(resBTNEXIT)
lblBegin.Caption = ResolveResString(resLBLBEGIN)
piclInstall.ToolTipText =
    ResolveResString(resBTNTOOLTIPBEGIN)

Caption = gstrTitle
lblInstallMsg.Caption =
    ResolveResString(If(gfForceUseDefDest,
        resSPECNODEST, resSPECDEST), "I1",
        gstrAppName)
lblDestDir.Caption = gstrDestDir

If gfForceUseDefDest Then
    'We are forced to use the default destination directory, so the
    'user will not be able to change it.
    fraDir.Visible = False

    'Close in the blank space on the form by moving the Exit
    button to 'where this frame
    'currently is, and adjusting the size of the form respectively
    Dim yAdjust As Integer
    yAdjust = cmdExit.Top - linTopOfExitButtonIfNoDestDir.Y1
    cmdExit.Top = cmdExit.Top - yAdjust
    Height = Height - yAdjust

    EtchedLine Me, fraDir.Left, cmdExit.Top - cmdExit.Height \ 2,
        fraDir.Width
Else
    EtchedLine Me, fraDir.Left, cmdExit.Top - cmdExit.Height \ 2,
        fraDir.Width

End If

CenterForm Me
End Sub
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode
    As Integer)

```

```

If UnloadMode <> 1 Then
    ExitSetup Me, gintRET_EXIT
    Cancel = 1
End If
End Sub

```

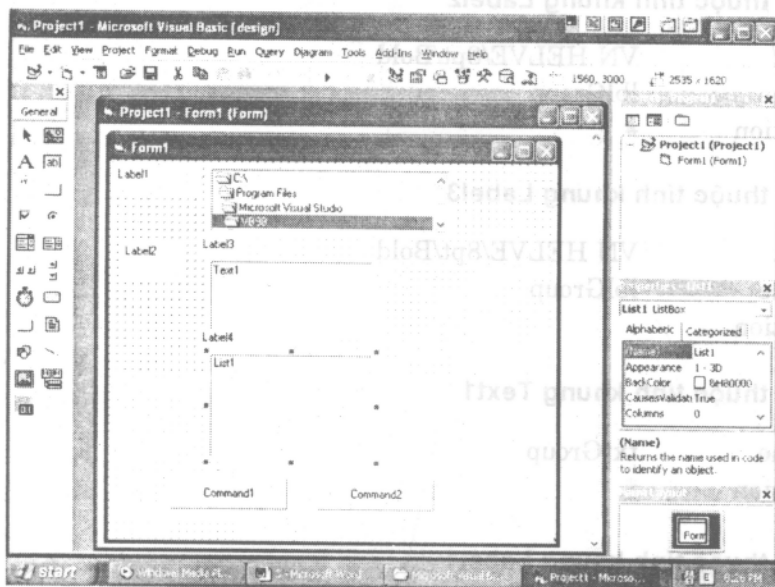
Bài 14: Tạo khung thoại kết nhóm đối tượng

Với bài tập mẫu này, bạn có thể thay đổi các thuộc tính (được minh họa bằng các dấu #) và các mã chương trình trong khung cửa sổ Code của chúng để thiết kế khung thoại theo ý muốn.

1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích công cụ Label trong thanh công cụ, chọn vị trí gần ngay trên cùng khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung nhãn Label1.
4. Kích công cụ ChDirBox trong thanh công cụ, chọn vị trí trong khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung tìm đối tượng chương trình.
5. Kích công cụ Label trong thanh công cụ, chọn vị trí, kích chìm nút mouse và kéo để tạo khung nhãn Label2.
6. Lặp lại bước 5 để tạo khung nhãn Label3.
7. Kích công cụ TextBox trong thanh công cụ, chọn vị trí trong khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung nhập ký tự.
8. Lặp lại bước 6 để tạo khung nhãn Label4.
9. Kích công cụ ListBox trong thanh công cụ, chọn vị trí trong khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung nhập danh sách.

10. Kích công cụ CommandButton trong thanh công cụ General. Kích và kéo mouse ngay vị trí bên trái gần cuối khung mẫu biểu Form để tạo khung nút lệnh Command1.
11. Lặp lại bước 10 để tạo khung nút lệnh Command2.

Mẫu biểu thiết kế ban đầu có dạng như sau:



Gán thuộc tính đối tượng

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	frmGroup
Caption	*#

Gán thuộc tính khung Label1

Font	VNI Helve/10pt/Bold
Name	lblMain

Alignment	Left
Caption	#

Gán thuộc tính khung ChDirBox

Name	dir95Group
------	------------

Gán thuộc tính khung Label2

Font	VN HELVE/8pt/Bold
Name	lblGroup
Caption	#

Gán thuộc tính khung Label3

Font	VN HELVE/8pt/Bold
Name	lblGroup
Caption	

Gán thuộc tính khung Text1

Name	txtGroup
Caption	*

Gán thuộc tính khung Label4

Font	VN HELVE/8pt/
Name	lblGroups
Caption	#

Gán thuộc tính khung ListBox

Font	VN HELVE/8pt/
Name	lstGroups
Caption	

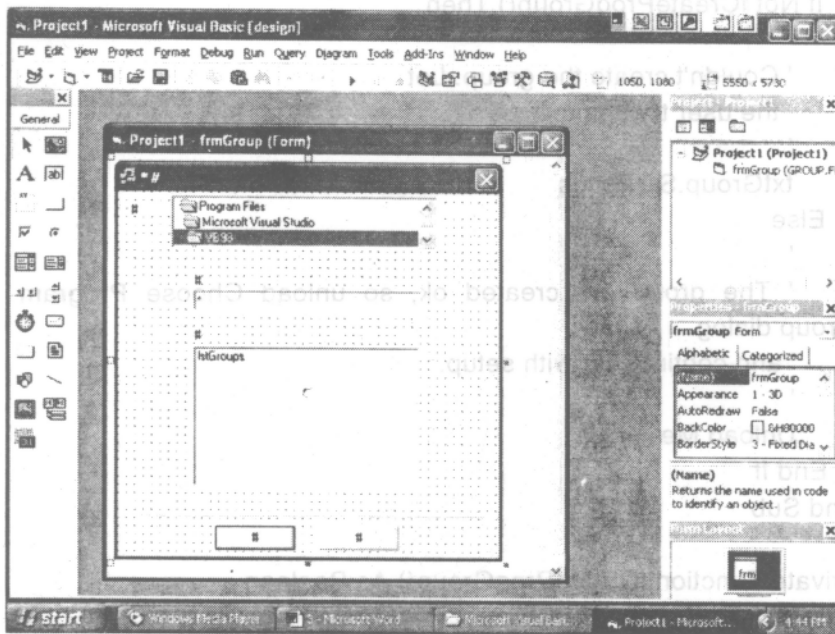
Gán thuộc tính nút lệnh Command1

Font	MS Sans Serif
------	---------------

Name cmdContinue
Caption #

Gán thuộc tính nút lệnh Command2

Font MS Sans Serif
Name cmdCancel
Caption #



Chương trình của mẫu biểu trong khung cửa sổ Code:

Option Explicit

Dim mstrGroup As String

Dim mstrDesktop As String

Dim mstrProgramsPath As String

Dim mfrm As Form

Dim fPrivate As Boolean

Dim fStartMenu As Boolean

```
Private Sub cmdCancel_Click()  
    ExitSetup frmGroup, gintRET_EXIT  
End Sub
```

```
Private Sub cmdContinue_Click()  
    mstrGroup = txtGroup.Text  
    If Not fCreateProgGroup() Then  
        ' Couldn't create the group. Let  
        ' the user try again.  
        txtGroup.SetFocus  
    Else  
        ' The group got created ok, so unload Choose Program  
        Group dialog  
        ' and continue on with setup.  
        Unload Me  
    End If  
End Sub
```

Private Function fCreateProgGroup() As Boolean

' Create a program group for either NT or Win95.

```
Dim strMsg As String  
If TreatAsWin95() Then  
    If Not fValidFilename(mstrGroup) Then  
        strMsg =  
        ResolveResString(resGROUPINVALIDGROUPNAME, "1",  
        CStr(gintMAX_PATH_LEN), "2",  
        ResolveResString(resCOMMON_INVALIDFILECHARS))
```

```

        MsgFunc strMsg, vbOKOnly Or vbQuestion, gstrTitle
        GoTo CGError
    End If
Else
    If Not fValidNTGroupName(mstrGroup) Then
        strMsg =
            ResolveResString(resGROUPINVALIDGROUP
            NAME, "I1",
            CStr(gintMAX_GROUPNAME_LEN%), "I2",
            ResolveResString(resGROUPINVALIDCHARS))
        MsgFunc strMsg, vbOKOnly Or vbQuestion, gstrTitle
        GoTo CGError
    End If

End If

'
'Go ahead and create the main program group
'
If Not fCreateOSProgramGroup(mfrm, mstrGroup, True, ,
                            fPrivate, fStartMenu) Then
    GoTo CGError
End If

fCreateProgGroup = True
Exit Function
CGError:
fCreateProgGroup = False
End Function

Private Sub Form_Load()
'
' Initialize localized control properties.
'

SetFormFont Me
Me.Caption = ResolveResString(resGROUPFRM, "I1",
                              gstrAppName)

```

```

lblMain.Caption = ResolveResString(resGROUPLBLMAIN)
lblGroup.Caption = ResolveResString(resGROUPLBLGROUP)
lblGroups.Caption =
ResolveResString(resGROUPLBLGROUPS)
cmdContinue.Caption =
ResolveResString(resGROUPBTNCONTINUE)
cmdCancel.Caption = ResolveResString(resLOG_vbCancel)
'
' Initialize the Program Group text box with the
' title of the application.
'
txtGroup.Text = gstrTitle
'
' Load the ListBox with the program manager groups.
'
If TreatAsWin95() Then
    LoadW95Groups
Else
    LoadProgManGroups
End If
'
' Initialize the Program Group textbox with the
' default group selected in the list box.
'
txtGroup.Text = lstGroups.List(lstGroups.ListIndex)
End Sub

Private Sub lstGroups_Click()
    txtGroup.Text = lstGroups.List(lstGroups.ListIndex)
End Sub

Private Sub txtGroup_Change()
    cmdContinue.Enabled = Len(Trim(txtGroup.Text)) > 0
End Sub
Sub LoadProgManGroups()

```

```

'
' This routine uses DDE to talk to Program Manager
' to retrieve a list of all the groups it manages.
' It should only be called if the shell is NT 3.51.
' If it is Win95 or NT4, call LoadW95Groups()
' instead.
'
' Special strings used in this routine. Do not
' localize these strings.
'

Const strPROGMANLINKTOPIC = "ProgMan\Progman"
Const strPROGMANLINKITEM = "Progman"
Const strNDWGROUPO = "Quick Access"

Dim strGroups As String
Dim strGroup As String
Dim intOffset As Integer
Dim intAnchor As Integer
Dim iGroup As Long

lbidDE.LinkTopic = strPROGMANLINKTOPIC
lbidDE.LinkItem = strPROGMANLINKITEM
lbidDE.LinkMode = 2
lbidDE.LinkRequest
On Error Resume Next
lbidDE.LinkMode = 0
'

' The DDE call just made put the names of all the groups
' into the caption property of the lbIDDE control.
' We want to transfer them to the list box. They are
' separated by vbCrLf's.
'

strGroups = lbIDDE.Caption
intAnchor = 1
intOffset = InStr(intAnchor, strGroups, vbCrLf)

```

```

lstGroups.Clear
Do While intOffset > 0
    strGroup = Mid(strGroups, intAnchor, intOffset - intAnchor)
    '
    ' Norton Desktop for Windows uses the "Quick Access" group
    ' to replace program manager. Trying to add icons to this
    ' group will fail later when we perform our DDE.linkrequest.
    ' Therefore, skip this group.
    '
    If strGroup <> strNDWGROUP Then
        lstGroups.AddItem strGroup
    End If

    intAnchor = intOffset + 2
    intOffset = InStr(intAnchor, strGroups, vbCrLf)
Loop
'
' The lstGroups listbox now contains a listing of all the program
' manager groups.
'
' Look for the default group in the list and select it. If it's
' not there, add it.
'
iGroup      =      SendMessageString(lstGroups.hWnd,
    LB_FINDSTRINGEXACT, -1, mstrDefGroup)
If iGroup = LB_ERR Then
    '
    ' The group doesn't yet exist, add it to the list.
    '
    lstGroups.AddItem mstrDefGroup
    lstGroups.ListIndex = lstGroups.NewIndex
Else
    lstGroups.ListIndex = iGroup
End If
End Sub

```



```

Sub LoadW95Groups()
'
' This routine uses the system registry to
' retrieve a list of all the subfolders in the
' \windows\start menu\programs folder.
' It should only be called if the shell is Win95
' NT4. If it is NT 3.51, call LoadProgManGroups()
' instead.
'
    Dim strFolder As String
    Dim iFolder As Integer

    mstrProgramsPath = strGetProgramsFilesPath()
    strFolder = Dir(mstrProgramsPath, vbDirectory) ' Retrieve the
                                                    first entry.
    lstGroups.Clear
    Do While strFolder <> ""
        ' Ignore the current directory and the encompassing directory.
        '
        If strFolder <> "." And strFolder <> ".." Then
            ' Verify that we actually got a directory and not a file.
            '
            If (GetAttr(mstrProgramsPath & strFolder) And
                vbDirectory) = vbDirectory Then
                ' We got a directory, add it to the list.
                '
                lstGroups.AddItem strFolder
            End If
        End If
        ' Get the next subfolder in the Programs folder
    
```

```

        strFolder = Dir
Loop
'
' The lstGroups listbox now contains a listing of all the Programs
' subfolders (the groups).
'
' Look for the default folder in the list and select it.  If it's
' not there, add it.
'
iFolder      =      SendMessageString(lstGroups.hWnd,
        LB_FINDSTRINGEXACT, -1, mstrDefGroup)
If iFolder = LB_ERR Then
'
' The group doesn't yet exist, add it to the list.
'
        lstGroups.AddItem mstrDefGroup
        lstGroups.ListIndex = lstGroups.NewIndex
Else
        lstGroups.ListIndex = iFolder
End If
End Sub

Public Property Get GroupName(frm As Form, strDefGroup As
String, Optional fPriv As Boolean = True, Optional ByVal fStart As
Boolean = False) As String
        mstrDefGroup = strDefGroup
        Set mfrm = frm

        fPrivate = fPriv
        fStartMenu = fStart
        If gfNoUserInput = True Then
                mstrGroup = mstrDefGroup
                If Not fCreateProgGroup() Then
                        ExitSetup frmSetup1, gintRET_FATAL
                End If

```

```

Else
    Me.Show vbModal
End If
GroupName = mstrGroup
End Property

Private Sub txtGroup_GotFocus()
    txtGroup.SelStart = 0
    txtGroup.SelLength = 32767
End Sub

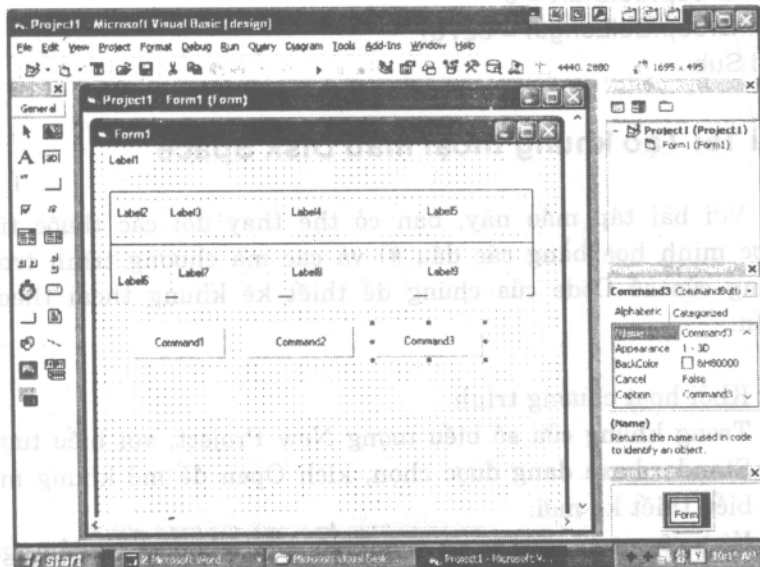
```

Bài 15: Tạo khung thoại mẫu Disk Space

Với bài tập mẫu này, bạn có thể thay đổi các thuộc tính (được minh họa bằng các dấu #) và các mã chương trình trong khung của số Code của chúng để thiết kế khung thoại theo ý muốn.

1. Khởi hoạt chương trình.
2. Trong khung của sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích công cụ Label trong thanh công cụ, chọn vị trí gần ngay trên cùng khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung nhân Label1.
4. Kích công cụ Shape trong thanh công cụ, chọn vị trí trong khung mẫu biểu Form, kích chìm nút mouse  và kéo để tạo khung.
5. Kích lại công cụ Label trong thanh công cụ, kích chìm nút mouse vào bên trong khung Shape và kéo để tạo khung nhân Label2.
6. Lập lại bước 5 để tạo thêm ba khung nhân còn lại.
7. Lập lại bước 4 để kẻ thêm một khung Shape2 nằm ngay dưới khung Shape thứ nhất.

8. Lặp lại bước 5 và 6 để tạo bốn khung nhãn nhãn trong khung Shape2.
9. Kích công cụ CommandButton trong thanh công cụ General. Kích và kéo mouse ngay vị trí bên trái gần cuối khung mẫu biểu Form để tạo khung nút lệnh Command1.
10. Lặp lại bước 10 để tạo thêm hai khung nút lệnh Command2 và Command3.



Gán thuộc tính đối tượng

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	frmDiskSpace
Caption	#

Gán thuộc tính khung Label1

Font	VNI Helve/10pt/Bold
------	---------------------

Name	lblNoSpace
Alignment	Left
Caption	#
BorderStyle	0-None
Caption	#

Gán thuộc tính khung Shape1

Name	ShpHeading
BackColor	màu trắng
BorderStyle	1-Solid
Shape	0-Rectangle

Gán thuộc tính khung Label2

Font	VN HELVE/8pt/Bold
Name	lblDiskH
Alignment	Left
BorderStyle	0-None
Caption	#

Gán thuộc tính khung Label3

Font	VN HELVE/8pt/Bold
Name	lblReqH
Alignment	Right
BorderStyle	0-None
Caption	#

Gán thuộc tính khung Label4

Font	VN HELVE/8pt/Bold
Name	lblAvailH
Alignment	Right
BorderStyle	0-None
Caption	#

Gán thuộc tính khung Label5

Font	VN HELVE/8pt/Bold
Name	lblNeedH
Alignment	Right
BorderStyle	0-None
Caption	#

Gán thuộc tính khung Shape2

Name	ShpHeading
BorderColor	màu đen
BorderStyle	1-Solid
Shape	0-Rectangle

Gán thuộc tính khung Label6

Font	VN HELVE/8pt/Bold
Name	lblDisk
Alignment	Left
BorderStyle	1-Fixed
Caption	

Gán thuộc tính khung Label7

Font	VN HELVE/8pt/Bold
Name	lblAvail
Alignment	Left
BorderStyle	1-Fixed
Caption	

Gán thuộc tính khung Label8

Font	VN HELVE/8pt/Bold
Name	lblReq
Alignment	Left
BorderStyle	1-Fixed
Caption	

Gán thuộc tính nút lệnh Command1

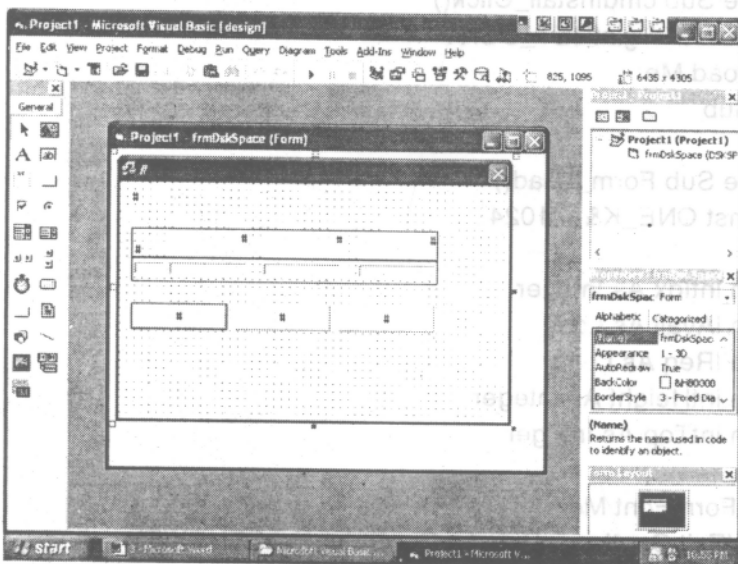
Font	MS Sans Serif
Name	cmdChgDrv
Caption	#

Gán thuộc tính nút lệnh Command2

Font	MS Sans Serif
Name	cmdInstall
Caption	#

Gán thuộc tính nút lệnh Command3

Font	MS Sans Serif
Name	cmdExit
Caption	#



Mã lệnh chương trình trong khung của sổ Code:

Option Explicit
Option Compare Text

Form/Module Constants

Const strFMT\$ = "#####0 K"

Private Sub cmdChgDrv_Click()
 gfRetVal = gintRET_CANCEL
 Unload Me
End Sub

Private Sub cmdExit_Click()
 ExitSetup Me, gintRET_EXIT
End Sub

Private Sub cmdInstall_Click()
 gfRetVal = gintRET_CONT
 Unload Me
End Sub

Private Sub Form_Load()
 Const ONE_K& = 1024

 Dim intIdx As Integer
 Dim lAvail As Long
 Dim lReq As Long
 Dim intHeight As Integer
 Dim intTop As Integer

 SetFormFont Me
 cmdExit.Caption = ResolveResString(resBTNEXIT)
 cmdInstall.Caption = ResolveResString(resBTNINSTALLNOW)
 cmdChgDrv.Caption = ResolveResString(resBTNCHGDRV)


```

lblDiskH.Caption = ResolveResString(resLBLDRIVE)
lblAvailH.Caption = ResolveResString(resLBLAVAIL)
lblNeedH.Caption = ResolveResString(resLBLNEEDED)
lblReqH.Caption = ResolveResString(resLBLREQUIRED)
lblNoSpace.Caption = ResolveResString(resLBLNOSPACE)
frmDskSpace.Caption = gstrTitle

```

```

intHeight = lblDisk(0).Height * 1.6
intTop = lblDisk(0).Top

```

```

' borders are for design mode only...

```

```

lblDisk(0).BorderStyle = 0
lblReq(0).BorderStyle = 0
lblAvail(0).BorderStyle = 0
lblNeed(0).BorderStyle = 0

```

```

For intIdx = 1 To Len(gstrDrivesUsed)

```

```

    Load lblDisk(intIdx)
    Load lblReq(intIdx)
    Load lblAvail(intIdx)
    Load lblNeed(intIdx)

```

```

    lAvail = gsDiskSpace(intIdx).lAvail
    lReq = gsDiskSpace(intIdx).lReq

```

```

    lblDisk(intIdx).Caption = Mid$(gstrDrivesUsed, intIdx, 1) &
                             gstrCOLON
    lblReq(intIdx).Caption = Format$(lReq / ONE_K, strFMT)
    lblAvail(intIdx).Caption = Format$(lAvail / ONE_K, strFMT)
    lblNeed(intIdx).Caption = Format$(If(lReq > lAvail, lReq -
                                         lAvail, 0) / ONE_K, strFMT)

```

```

    lblDisk(intIdx).Top = intTop

```

```

lblReq(intIdx).Top = intTop
lblAvail(intIdx).Top = intTop
lblNeed(intIdx).Top = intTop

```

```

intTop = intTop + intHeight

```

```

lblDisk(intIdx).Visible = True
lblReq(intIdx).Visible = True
lblAvail(intIdx).Visible = True
lblNeed(intIdx).Visible = True

```

```

Next

```

```

shpSpace.Height = intHeight * (intIdx - 1)
cmdChgDrv.Top = shpSpace.Top + shpSpace.Height +
                cmdChgDrv.Height
cmdInstall.Top = cmdChgDrv.Top
cmdExit.Top = cmdChgDrv.Top

```

```

frmDskSpace.Height = cmdChgDrv.Top + cmdChgDrv.Height *
2.5

```

```

EtchedLine Me, 100, cmdChgDrv.Top - cmdChgDrv.Height *
0.5, Me.ScaleWidth - 200

```

```

CenterForm Me
End Sub

```

```

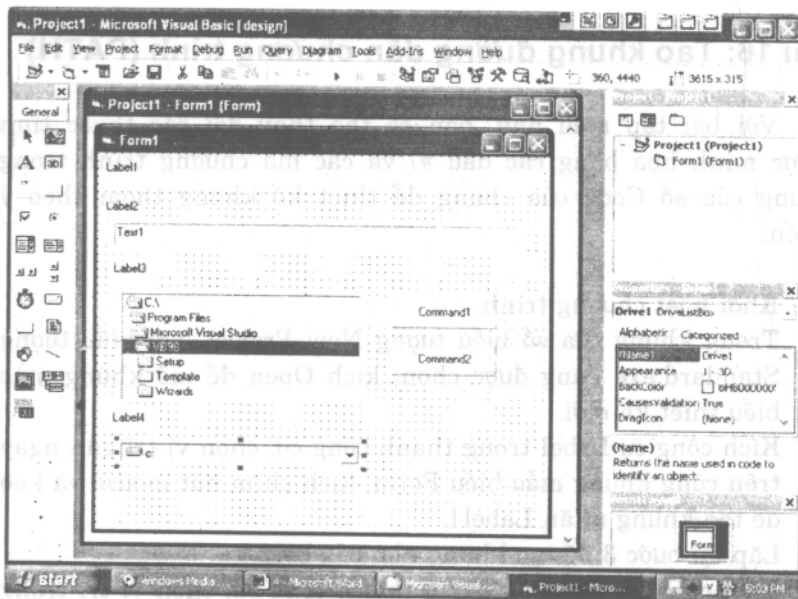
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode
As Integer)
    If UnloadMode <> 1 Then
        ExitSetup Me, gintRET_EXIT
        Cancel = 1
    End If
End Sub

```

Bài 16: Tạo khung đường dẫn chương trình (PATH)

Với bài tập mẫu này, bạn có thể thay đổi các thuộc tính (được minh họa bằng các dấu #) và các mã chương trình trong khung cửa sổ Code của chúng để thiết kế khung thoại theo ý muốn.

1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích công cụ Label trong thanh công cụ, chọn vị trí gần ngay trên cùng khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung nhãn Label1.
4. Lập lại bước 3 để tạo khung nhãn Label2.
5. Kích công cụ TextBox trong thanh công cụ, chọn vị trí trong khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung ký tự.
6. Lập lại bước 4 để tạo khung nhãn Label3.
7. Kích công cụ ChkDirList trong thanh công cụ, chọn vị trí dưới khung nhãn Label3, kích chìm nút mouse  và kéo để tạo danh sách chương trình.
8. Kích công cụ CommandButton trong thanh công cụ General. Kích và kéo mouse ngay vị trí bên phải khung ChkDirList để tạo khung nút lệnh Command1.
9. Lập lại bước 8 để tạo thêm khung nút lệnh Command2.
10. Lập lại bước 6 để tạo khung nhãn Label4.
11. Kích công cụ DirDriver trong thanh công cụ, chọn vị trí dưới khung nhãn Label4, kích chìm nút mouse  và kéo để tạo khung ổ đĩa.



Gán thuộc tính đối tượng

Đặt tên mẫu biểu

Font MS Sans Serif
Name frmPath
Caption #

Gán thuộc tính khung Label1

Font VNI Helve/10pt/Bold
Name lblNolblPromt
Alignment Left
BorderStyle 0-None
Caption *****

Gán thuộc tính khung Label2

Font VN HELVE/8pt/Bold

Name	lblPath
Alignment	Left
BorderStyle	0-None
Caption	#

Gán thuộc tính khung TextBox

Font	VN HELVE/8pt/Bold
Name	lbltxtBox
Alignment	Left
Caption	#

Gán thuộc tính khung Label3

Font	VN HELVE/8pt/Bold
Name	lblDirs
Alignment	Left
BorderStyle	0-None
Caption	#

Gán thuộc tính khung ChkDirBox

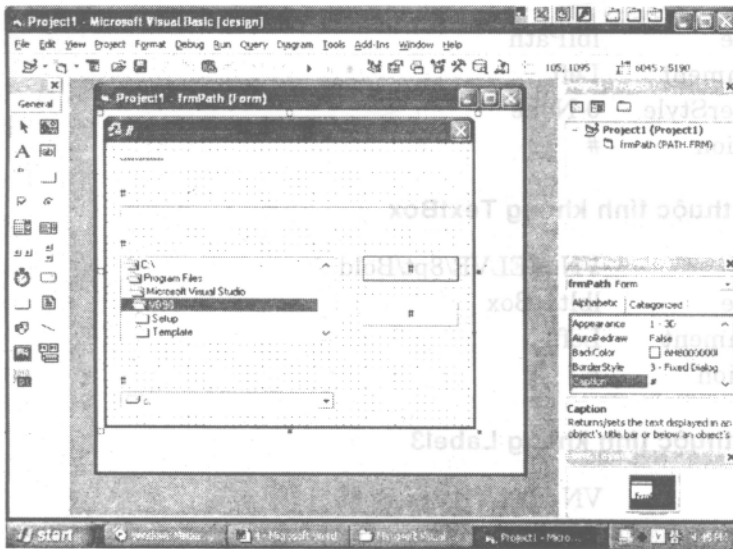
Font	VN HELVE/8pt/Bold
Name	dirDirs

Gán thuộc tính nút lệnh Command1

Font	MS Sans Serif
Name	cmdChgDrv
Caption	#

Gán thuộc tính nút lệnh Command2

Font	MS Sans Serif
Name	cmdInstall
Caption	#



Mã lệnh trong khung cửa sổ Code có dạng:

Option Explicit

Option Compare Text

' Form/Module Variables

Dim mfMustExist As Integer

Dim mfCancelExit As Integer

Private Sub cmdCancel_Click()

 If mfCancelExit = True Then

 ExitSetup Me, gintRET_EXIT

 Else

 gfRetVal = gintRET_CANCEL

 Unload Me

 End If

End Sub

```

Private Sub cmdOK_Click()
    Dim strPathName As String
    Dim strMsg As String
    Dim intRet As Integer

    SetMousePtr vbHourglass

    strPathName = ResolveDir(txtPath.Text, mfMustExist, True)

    If strPathName <> vbNullString Then
        If frmSetup1.Tag = gstrDIR_DEST And strPathName <>
gstrDestDir Then
            If DirExists(strPathName) = False Then
                strMsg = ResolveResString(resDESTDIR) & vbCrLf & vbCrLf
                    & strPathName
                strMsg = strMsg & vbCrLf & vbCrLf &
                    ResolveResString(resCREATE)
                intRet = MsgFunc(strMsg, vbYesNo Or vbQuestion,
gstrTitle)
                If gfNoUserInput = True Then
                    ExitSetup Me, gintRET_FATAL
                End If
                If intRet = vbNo Then txtPath.SetFocus
                    SetMousePtr gintMOUSE_DEFAULT
                    Exit Sub
                End If
            End If
            If IsValidDestDir(strPathName) = False Then
                txtPath.SetFocus
                SetMousePtr gintMOUSE_DEFAULT
                Exit Sub
            End If
        End If

        frmSetup1.Tag = strPathName
    End If

```

```

        gfRetVal = gintRET_CONT
        Unload Me
    Else
        txtPath.SetFocus
    End If

    SetMousePtr gintMOUSE_DEFAULT
End Sub

Private Sub dirDirs_Change()
    Static intBusy As Integer

    On Error Resume Next

    If intBusy = False Then
        intBusy = True

        ChDir dirDirs.Path

        If Err = 0 Then
            txtPath.Text = dirDirs.Path
            drvDrives.Drive = Left$(dirDirs.Path, 2)
        Else
            Err = 0
        End If

        intBusy = False
    End If
End Sub

Private Sub drvDrives_Change()
    Static strOldDrive As String
    Static intBusy As Integer

    Dim strDrive As String

```



```

If intBusy = False Then
    intBusy = True

    strDrive = drvDrives.Drive

    If CheckDrive(strDrive, Me.Caption) = True Then
        strOldDrive = strDrive
        dirDirs.Path = strDrive
    Else
        drvDrives.Drive = strOldDrive
    End If

    intBusy = False
End If
End Sub

Private Sub Form_Load()
    On Error Resume Next

    SetMousePtr vbHourglass

    SetFormFont Me
    cmdOK.Caption = ResolveResString(resBTNOK)
    lblDrives.Caption = ResolveResString(resLBLDRIVES)
    lblDirs.Caption = ResolveResString(resLBLDIRS)
    lblPath.Caption = ResolveResString(resLBLPATH)

    If frmSetup1.Tag = gstrDIR_SRC Then
        Caption = ResolveResString(resINSTFROM)
        lblPrompt.Caption = ResolveResString(resSRCPROMPT,
            "I1", gstrAppName)
        cmdCancel.Caption = ResolveResString(resBTNEXIT, "I1",
            gstrAppName)
        mfCancelExit = True
        dirDirs.Path = gstrSrcPath
    End If
End Sub

```

```

If Err > 0 Then
    dirDirs.Path = Left$(App.Path, 3)
End If
mfMustExist = True
Else
    Caption = ResolveResString(resCHANGEDIR)
    lblPrompt.Caption = ResolveResString(resDESTPROMPT)
    cmdCancel.Caption = ResolveResString(resBTNCANCEL)
    mfCancelExit = False
    dirDirs.Path = gstrDestDir
    If Err > 0 Then
        'Next try root of destination drive
        If Len(gstrDestDir) >= 2 Then
            If Mid$(gstrDestDir, 2, 1) = gstrCOLON Then
                Err = 0
                dirDirs.Path = Left$(gstrDestDir, 2) & gstrSEP_DIR
            End If
        End If
    End If
    If Err > 0 Then
        dirDirs.Path = Left$(App.Path, 3)
    End If

    'Init txtPath.Text to gstrDestDir even if this
    ' directory does not (yet) exist.
    txtPath.Text = gstrDestDir
    mfMustExist = False
End If

If frmSetup1.Tag = gstrDIR_SRC Then
    txtPath.Text = dirDirs.Path
End If

drvDrives.Drive = Left$(dirDirs.Path, 2)
drvDrives_Change

```

```
SetMousePtr gintMOUSE_DEFAULT
```

```
CenterForm Me
```

```
'Highlight all of txtPath's text so that typing immediately  
overwrites it
```

```
txtPath.SelStart = 0
```

```
txtPath.SelLength = Len(txtPath.Text)
```

```
Err = 0
```

```
End Sub
```

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode  
As Integer)
```

```
If UnloadMode <> 1 Then
```

```
If mfCancelExit = True Then
```

```
ExitSetup Me, gintRET_EXIT
```

```
Cancel = 1
```

```
Else
```

```
gfRetVal = gintRET_CANCEL
```

```
Unload Me
```

```
End If
```

```
End If
```

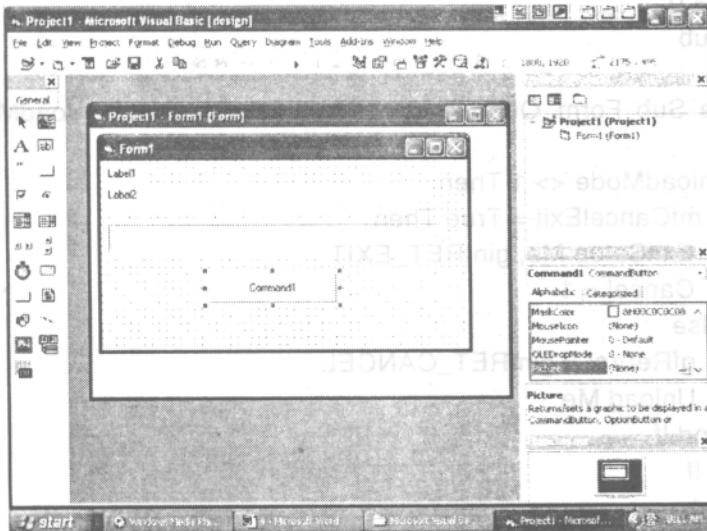
```
End Sub
```

Bài 17: Tạo khung màn hình Copy

Trong bài thực hành, bạn thiết kế một khung thoại hiện tình trạng sao chép tập tin từ vị trí này sang vị trí khác.

1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, với biểu tượng Standard.exe đang được chọn, kích Open để mở khung mẫu biểu thiết kế mới.
3. Kích công cụ Label trong thanh công cụ, chọn vị trí gần ngay trên cùng khung mẫu biểu Form, kích chìm nút mouse và kéo

- để tạo khung nhãn Label1.
4. Lập lại bước 3 để tạo khung nhãn Label2.
5. Kích công cụ PictureBox trong thanh công cụ, chọn vị trí trong khung mẫu biểu Form, kích chìm nút mouse và kéo để tạo khung hình trạng.
6. Kích công cụ CommandButton trong thanh công cụ General. Kích và kéo mouse ngay dưới khung PictureBox để tạo khung nút lệnh Command.



Gán thuộc tính đối tượng

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	frmCopy
Caption	COPY

Gán thuộc tính khung Label1

Font	VNI Helve/10pt/Bold
------	---------------------

Name	lblCopy
Alignment	Left
BorderStyle	0-None
Caption	A:\

Gán thuộc tính khung Label2

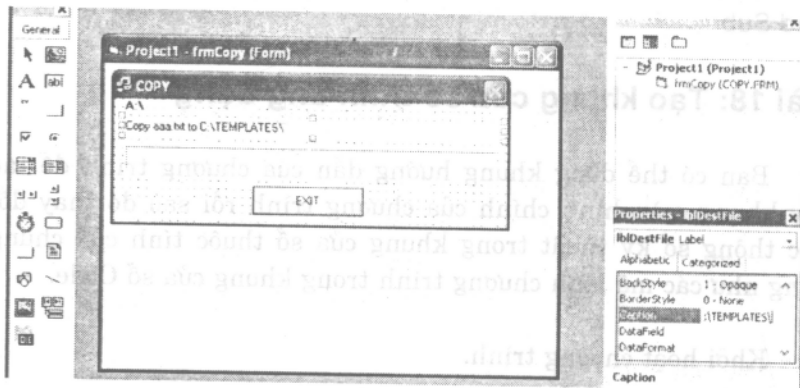
Font	VN HELVE/8pt
Name	lblDestFile
Alignment	Left
BorderStyle	0-None
Caption	Copy aaa.txt to C:\TEMPLATES\

Gán thuộc tính khung PictureBox

Font	VN HELVE/8pt/Bold
Name	lblpicStatus
BorderStyle	1-Fixed Single
FillColor	màu xanh

Gán thuộc tính nút lệnh Command

Font	MS Sans Serif
Name	cmdExit
Caption	EXIT



Mã chương trình có dạng:

```
Option Explicit
Option Compare Text
' frmCopy
Private Sub cmdExit_Click()
    ExitSetup Me, gintRET_EXIT
End Sub

Private Sub Form_Load()
    SetFormFont Me
    cmdExit.Caption = ResolveResString(resBTNCANCEL)
    lblCopy.Caption = ResolveResString(resLBLDESTFILE)
    lblDestFile.Caption = vbNullString

    frmCopy.Caption = gstrTitle
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode
    As Integer)
    If UnloadMode <> 1 Then
        ExitSetup Me, gintRET_EXIT
        Cancel = 1
    End If
End Sub
```

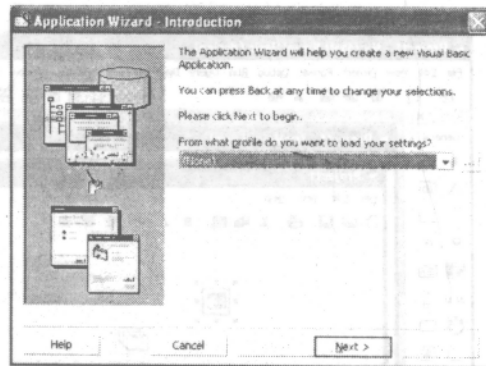
Bài 18: Tạo khung cửa sổ trình ứng dụng

Bạn có thể dùng khung hướng dẫn của chương trình để tạo lấy khung màn hình chính của chương trình rồi sau đó thay đổi các thông số kỹ thuật trong khung cửa sổ thuộc tính của chúng cũng như các mã lệnh chương trình trong khung cửa sổ Code.

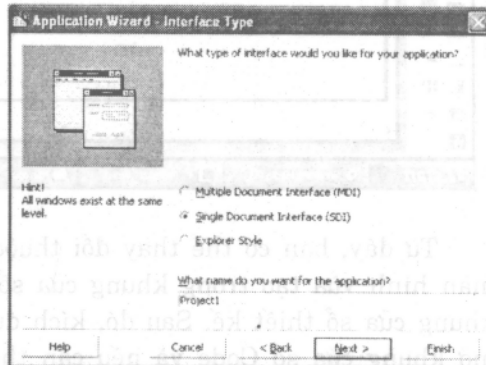
1. Khởi hoạt chương trình.

2. Trong khung cửa sổ biểu tượng New Project, kích biểu tượng VB Application Wizard, kích Open để mở khung hướng dẫn Application Wizard tạo khung cửa sổ chương trình.

3. Trong khung cửa sổ Application Wizard, nếu muốn dùng nền chương trình mới, kích ký hiệu ba chấm (...) nằm trong khung thông tin yêu cầu để mở khung thoại, nếu không, kích Next để chuyển sang khung màn hình hướng dẫn kế tiếp.

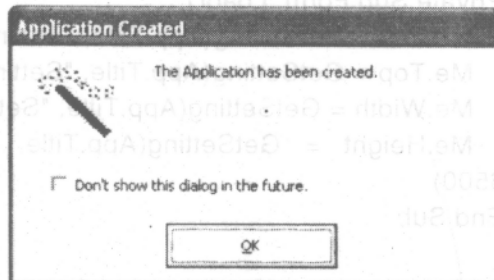


4. Trong khung màn hình hướng dẫn này, bạn có thể tùy chọn phần giao diện theo tác vụ. Trong thí dụ này, kích chọn thành phần giao diện đơn (SDI) và đặt tên cho trình ứng dụng.

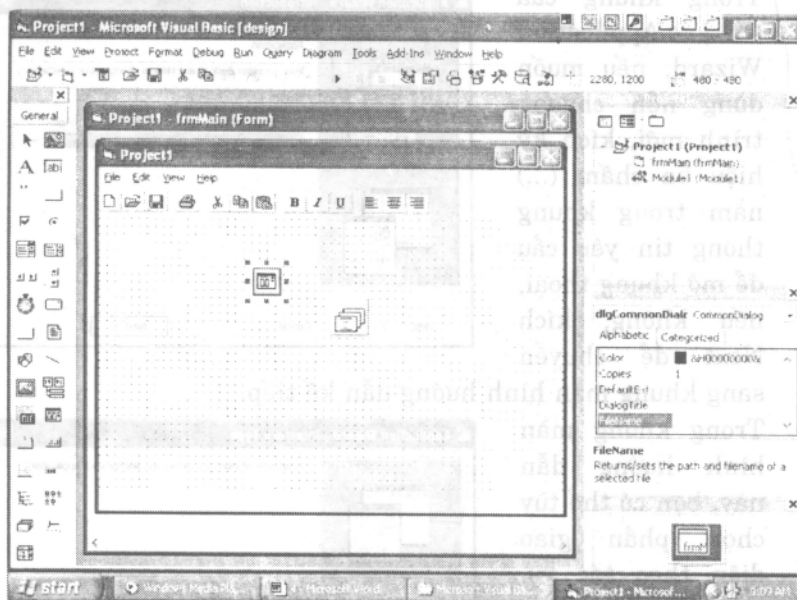


5. Sau khi đặt tên, có hai phương án để tạo khung màn hình ứng dụng:

Nếu muốn chấp nhận tất cả đối tượng mặc định của Wizard, kích Finish. Màn hình hiện khung thông tin



cho biết là khung màn hình trình ứng dụng đã được tạo và trong giấy lát, khung màn hình chính của chương trình vừa được thiết kế hiện lên màn hình.



Từ đây, bạn có thể thay đổi thuộc tính, tham số của khung màn hình vừa tạo trong khung của sổ Properties nằm bên phải khung của sổ thiết kế. Sau đó, kích đúp vào khung mẫu biểu để mở khung của sổ Code và nếu cần thiết, thay đổi nội dung mã lệnh của chương trình:

```
Private Sub Form_Load()
```

```
Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
```

```
Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
```

```
Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
```

```
Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
```

```
End Sub
```



```

Private Sub Form_Unload(Cancel As Integer)
    Dim i As Integer
    'close all sub forms
    For i = Forms.Count - 1 To 1 Step -1
        Unload Forms(i)
    Next
    If Me.WindowState <> vbMinimized Then
        SaveSetting App.Title, "Settings", "MainLeft", Me.Left
        SaveSetting App.Title, "Settings", "MainTop", Me.Top
        SaveSetting App.Title, "Settings", "MainWidth", Me.Width
        SaveSetting App.Title, "Settings", "MainHeight", Me.Height
    End If
End Sub

Private Sub tbToolBar_ButtonClick(ByVal Button As
    MSCoMctlLib.Button)
    On Error Resume Next
    Select Case Button.Key
        Case "New"
            'ToDo: Add 'New' button code.
            MsgBox "Add 'New' button code."
        Case "Open"
            mnuFileOpen_Click
        Case "Save"
            mnuFileSave_Click
        Case "Print"
            mnuFilePrint_Click
        Case "Cut"
            mnuEditCut_Click
        Case "Copy"
            mnuEditCopy_Click
        Case "Paste"
            mnuEditPaste_Click
        Case "Bold"

```

```

        'ToDo: Add 'Bold' button code.
        MsgBox "Add 'Bold' button code."
    Case "Italic"
        'ToDo: Add 'Italic' button code.
        MsgBox "Add 'Italic' button code."
    Case "Underline"
        'ToDo: Add 'Underline' button code.
        MsgBox "Add 'Underline' button code."
    Case "Align Left"
        'ToDo: Add 'Align Left' button code.
        MsgBox "Add 'Align Left' button code."
    Case "Center"
        'ToDo: Add 'Center' button code.
        MsgBox "Add 'Center' button code."
    Case "Align Right"
        'ToDo: Add 'Align Right' button code.
        MsgBox "Add 'Align Right' button code."
    End Select
End Sub

Private Sub mnuHelpAbout_Click()
    MsgBox "Version " & App.Major & "." & App.Minor & "." &
App.Revision
End Sub

Private Sub mnuHelpSearchForHelpOn_Click()
    Dim nRet As Integer

    'if there is no helpfile for this project display a message to the
user
    'you can set the HelpFile for your application in the
'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help
associated with this project.", vbInformation, Me.Caption
    End If
End Sub

```

```

Else
    On Error Resume Next
    nRet = OSWinHelp(Me.hwnd, App.HelpFile, 261, 0)
    If Err Then
        MsgBox Err.Description
    End If
End If

End Sub

Private Sub mnuHelpContents_Click()
    Dim nRet As Integer

    'if there is no helpfile for this project display a message to the
    user
    'you can set the HelpFile for your application in the
    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help
        associated with this project.", vbInformation, Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 3, 0)
        If Err Then
            MsgBox Err.Description
        End If
    End If

End Sub

End Sub

Private Sub mnuViewWebBrowser_Click()
    'ToDo: Add 'mnuViewWebBrowser_Click' code.
    MsgBox "Add 'mnuViewWebBrowser_Click' code."
End Sub

```

```

Private Sub mnuViewOptions_Click()
    'ToDo: Add 'mnuViewOptions_Click' code.
    MsgBox "Add 'mnuViewOptions_Click' code."
End Sub

Private Sub mnuViewRefresh_Click()
    'ToDo: Add 'mnuViewRefresh_Click' code.
    MsgBox "Add 'mnuViewRefresh_Click' code."
End Sub

Private Sub mnuViewStatusBar_Click()
    mnuViewStatusBar.Checked = Not mnuViewStatusBar.Checked
    mnuViewStatusBar.Checked = Not mnuViewStatusBar.Checked
    sbStatusBar.Visible = mnuViewStatusBar.Checked
End Sub

Private Sub mnuViewToolBar_Click()
    mnuViewToolBar.Checked = Not mnuViewToolBar.Checked
    tbToolBar.Visible = mnuViewToolBar.Checked
End Sub

Private Sub mnuEditPasteSpecial_Click()
    'ToDo: Add 'mnuEditPasteSpecial_Click' code.
    MsgBox "Add 'mnuEditPasteSpecial_Click' code."
End Sub

Private Sub mnuEditPaste_Click()
    'ToDo: Add 'mnuEditPaste_Click' code.
    MsgBox "Add 'mnuEditPaste_Click' code."
End Sub

Private Sub mnuEditCopy_Click()
    'ToDo: Add 'mnuEditCopy_Click' code.
    MsgBox "Add 'mnuEditCopy_Click' code."
End Sub

```

```

Private Sub mnuEditCut_Click()
    'ToDo: Add 'mnuEditCut_Click' code.
    MsgBox "Add 'mnuEditCut_Click' code."
End Sub

Private Sub mnuEditUndo_Click()
    'ToDo: Add 'mnuEditUndo_Click' code.
    MsgBox "Add 'mnuEditUndo_Click' code."
End Sub

Private Sub mnuFileExit_Click()
    'unload the form
    Unload Me
End Sub

Private Sub mnuFileSend_Click()
    'ToDo: Add 'mnuFileSend_Click' code.
    MsgBox "Add 'mnuFileSend_Click' code."
End Sub

Private Sub mnuFilePrint_Click()
    'ToDo: Add 'mnuFilePrint_Click' code.
    MsgBox "Add 'mnuFilePrint_Click' code."
End Sub

Private Sub mnuFilePrintPreview_Click()
    'ToDo: Add 'mnuFilePrintPreview_Click' code.
    MsgBox "Add 'mnuFilePrintPreview_Click' code."
End Sub

Private Sub mnuFilePageSetup_Click()
    On Error Resume Next
    With dlgCommonDialog
        .DialogTitle = "Page Setup"
        .CancelError = True
    End With
End Sub

```

```

        .ShowPrinter
    End With

End Sub

Private Sub mnuFileProperties_Click()
    'ToDo: Add 'mnuFileProperties_Click' code.
    MsgBox "Add 'mnuFileProperties_Click' code."
End Sub

Private Sub mnuFileSaveAll_Click()
    'ToDo: Add 'mnuFileSaveAll_Click' code.
    MsgBox "Add 'mnuFileSaveAll_Click' code."
End Sub

Private Sub mnuFileSaveAs_Click()
    'ToDo: Add 'mnuFileSaveAs_Click' code.
    MsgBox "Add 'mnuFileSaveAs_Click' code."
End Sub

Private Sub mnuFileSave_Click()
    'ToDo: Add 'mnuFileSave_Click' code.
    MsgBox "Add 'mnuFileSave_Click' code."
End Sub

Private Sub mnuFileClose_Click()
    'ToDo: Add 'mnuFileClose_Click' code.
    MsgBox "Add 'mnuFileClose_Click' code."
End Sub

Private Sub mnuFileOpen_Click()
    Dim sFile As String
    With dlgCommonDialog
        .DialogTitle = "Open"
        .CancelError = False
    End With

```

'ToDo: set the flags and attributes of the common dialog control

```
.Filter = "All Files (*.*)|*.*"
```

```
.ShowOpen
```

```
If Len(.FileName) = 0 Then
```

```
Exit Sub
```

```
End If
```

```
sFile = .FileName
```

```
End With
```

'ToDo: add code to process the opened file

End Sub

Private Sub mnuFileNew_Click()

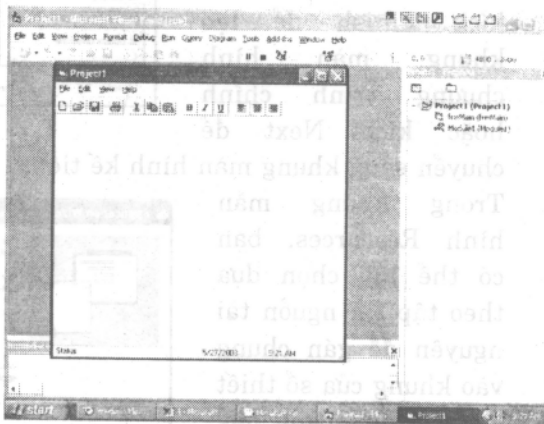
'ToDo: Add 'mnuFileNew_Click' code.

MsgBox "Add 'mnuFileNew_Click' code."

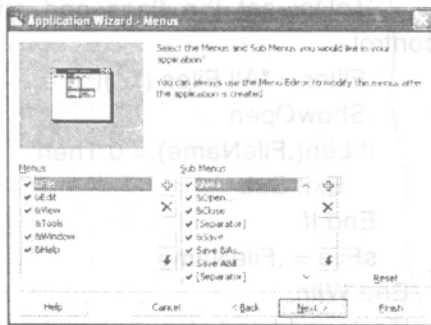
End Sub

Sau khi gõ rồi với lệnh Debug, ấn phím F5 để điều hoạt chương trình, và sau đó lưu lại tập tin thiết kế với tên tùy chọn; hoặc nếu sau khi quan sát, muốn thay đổi các công cụ hoặc các lệnh menu, ngay bước

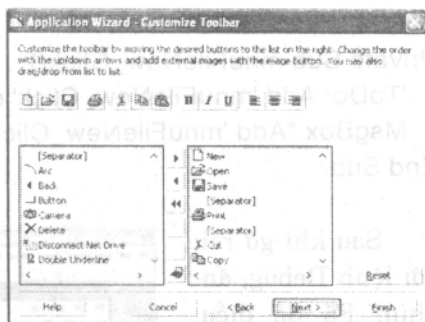
bước 4, kích Next để chuyển sang khung màn hình kế tiếp.



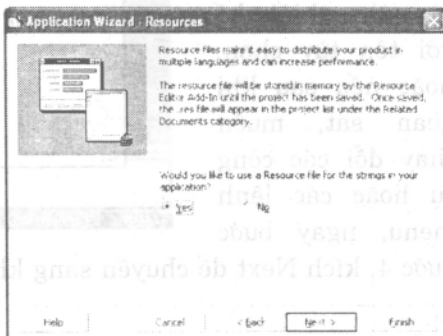
6. Ngay trong khung cửa sổ Menu, bạn có thể kích chọn hoặc xóa các thành phần menu chính cũng như menu con muốn hiện trong khung cửa sổ chương trình. Sau khi kết thúc, bạn có thể kích Finish để tạo khung màn hình chương trình chính hoặc kích Next để chuyển sang khung màn hình kế tiếp.



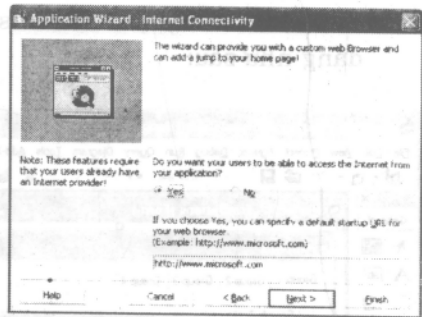
7. Trong khung cửa sổ Custom Toolbar, kích chọn thêm hoặc bỏ bớt các hộp công cụ cho phù hợp nhu cầu tác vụ. Sau khi kết thúc, bạn có thể kích Finish để tạo khung màn hình chương trình chính hoặc kích Next để chuyển sang khung màn hình kế tiếp.



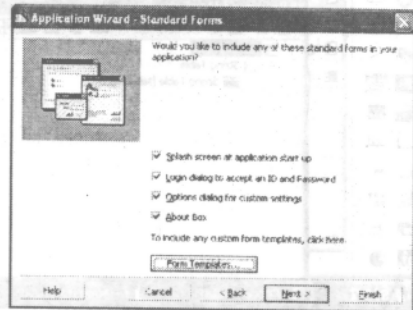
8. Trong khung màn hình Resources, bạn có thể tùy chọn dựa theo tập tin nguồn tài nguyên để gán chung vào khung cửa sổ thiết kế chương trình hay không. Sau khi tùy chọn, kích Next để chuyển sang khung màn hình kế tiếp, hoặc kích Finish để tạo khung màn hình chương trình chính.



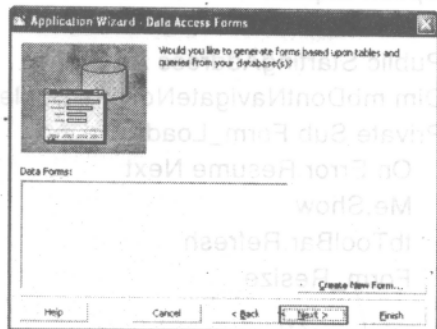
9. Trong khung cửa sổ kết nối, bạn có thể tùy chọn thành phần cho phép người sử dụng truy cập nội dung trình ứng dụng của bạn hay không. Sau khi kết thúc, bạn có thể kích Finish để tạo khung màn hình chương trình chính hoặc kích Next để chuyển sang khung màn hình kế tiếp.



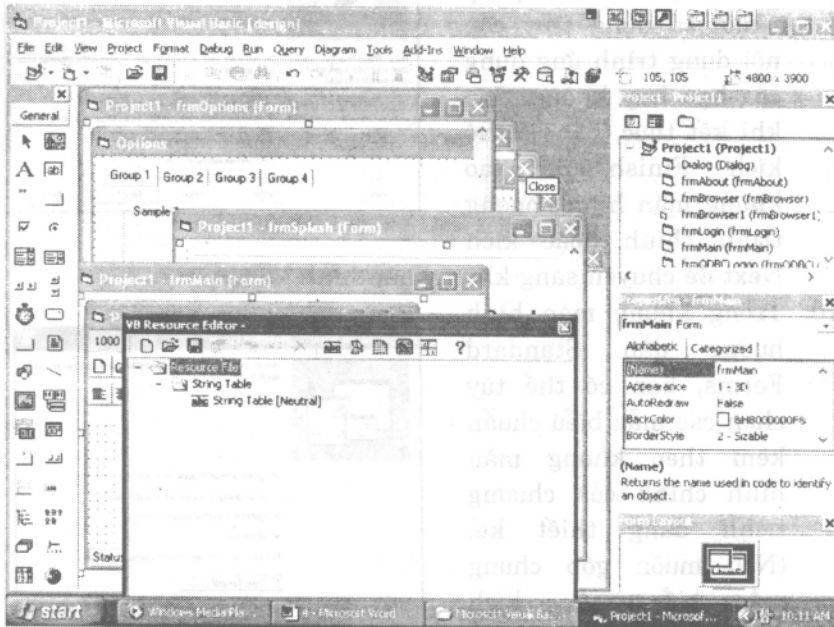
10. Trong khung màn hình hướng dẫn Standard Forms, bạn có thể tùy chọn các mẫu biểu chuẩn kèm theo khung màn hình chính của chương trình đang thiết kế. (Nếu muốn gộp chung mẫu biểu mẫu, kích thước lệnh Form Templates và tùy chọn đối tượng trong khung thoại hiện ra kế tiếp). Sau khi kết thúc, bạn có thể kích Finish để tạo khung màn hình chương trình chính hoặc kích Next để chuyển sang khung màn hình kế tiếp.



11. Trong khung cửa sổ Data Access Forms, bạn có thể tạo thêm mẫu biểu mới với nút lệnh Create New Form để tạo mẫu biểu truy cập dữ liệu và sau khi tạo mẫu biểu mới, tên mẫu biểu sẽ hiện trong khung Data Forms.



12. Sau khi kết thúc, kích Finish. Mẫu biểu thiết kế của bạn có dạng như sau:



Bạn có thể kích đúp vào từng mẫu biểu để quan sát mã lệnh chương trình và thay đổi theo nhu cầu cho phù hợp.

Option Explicit

```
Public StartingAddress As String
Dim mbDontNavigateNow As Boolean
Private Sub Form_Load()
    On Error Resume Next
    Me.Show
    tbToolBar.Refresh
    Form_Resize
End Sub
```

```
cboAddress.Move 50, lblAddress.Top + lblAddress.Height + 15
```

```
If Len(StartingAddress) > 0 Then  
    cboAddress.Text = StartingAddress  
    cboAddress.AddItem cboAddress.Text  
    'try to navigate to the starting address  
    tmTimer.Enabled = True  
    brwWebBrowser.Navigate StartingAddress  
End If
```

```
End Sub
```

```
Private Sub brwWebBrowser_DownloadComplete()  
    On Error Resume Next  
    Me.Caption = brwWebBrowser.LocationName  
End Sub
```

```
Private Sub brwWebBrowser_NavigateComplete(ByVal URL As  
String)  
    Dim i As Integer  
    Dim bFound As Boolean  
    Me.Caption = brwWebBrowser.LocationName  
    For i = 0 To cboAddress.ListCount - 1  
        If cboAddress.List(i) = brwWebBrowser.LocationURL Then  
            bFound = True  
            Exit For  
        End If  
    Next i  
    mbDontNavigateNow = True  
    If bFound Then  
        cboAddress.RemoveItem i  
    End If  
    cboAddress.AddItem brwWebBrowser.LocationURL, 0  
    cboAddress.ListIndex = 0  
    mbDontNavigateNow = False
```

End Sub

Private Sub cboAddress_Click()

If mbDontNavigateNow Then Exit Sub

timTimer.Enabled = True

brwWebBrowser.Navigate cboAddress.Text

End Sub

Private Sub cboAddress_KeyPress(KeyAscii As Integer)

On Error Resume Next

If KeyAscii = vbKeyReturn Then

cboAddress_Click

End If

End Sub

Private Sub Form_Resize()

cboAddress.Width = Me.ScaleWidth - 100

brwWebBrowser.Width = Me.ScaleWidth - 100

**brwWebBrowser.Height = Me.ScaleHeight - (picAddress.Top +
 picAddress.Height) - 100**

End Sub

Private Sub timTimer_Timer()

If brwWebBrowser.Busy = False Then

timTimer.Enabled = False

Me.Caption = brwWebBrowser.LocationName

Else

Me.Caption = "Working..."

End If

End Sub

Private Sub tbToolBar_ButtonClick(ByVal Button As Button)

On Error Resume Next

timTimer.Enabled = True

Select Case Button.Key

Case "Back"

brwWebBrowser.GoBack

Case "Forward"

brwWebBrowser.GoForward

Case "Refresh"

brwWebBrowser.Refresh

Case "Home"

brwWebBrowser.GoHome

Case "Search"

brwWebBrowser.GoSearch

Case "Stop"

timTimer.Enabled = False

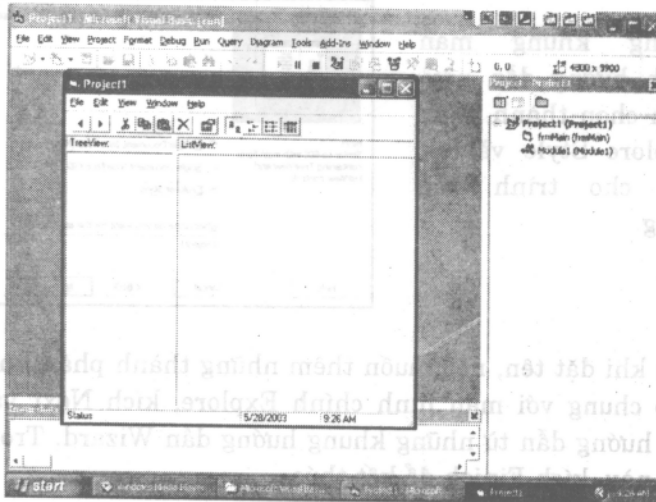
brwWebBrowser.Stop

Me.Caption = brwWebBrowser.LocationName

End Select

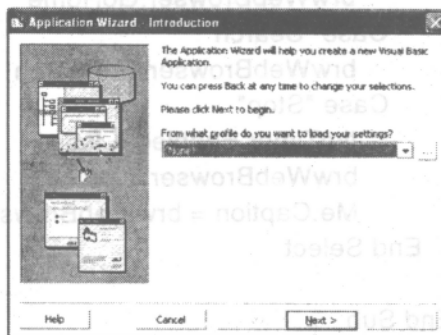
End Sub

Bài 19: Tạo khung màn hình Explorer

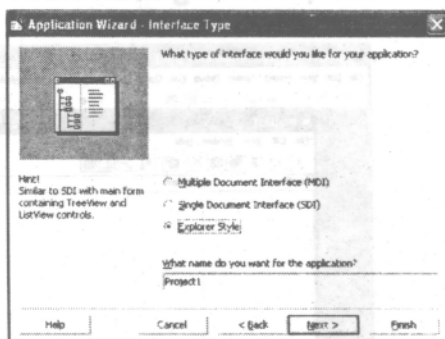


Bạn có thể dùng thành phần tiện ích VB Application Wizard để tạo khung màn hình Explorer và thay đổi một số thành phần đối tượng trong các khung hướng dẫn Wizard cũng như các mã lệnh trong khung cửa sổ Code để sửa đổi lại theo nhu cầu tác vụ.

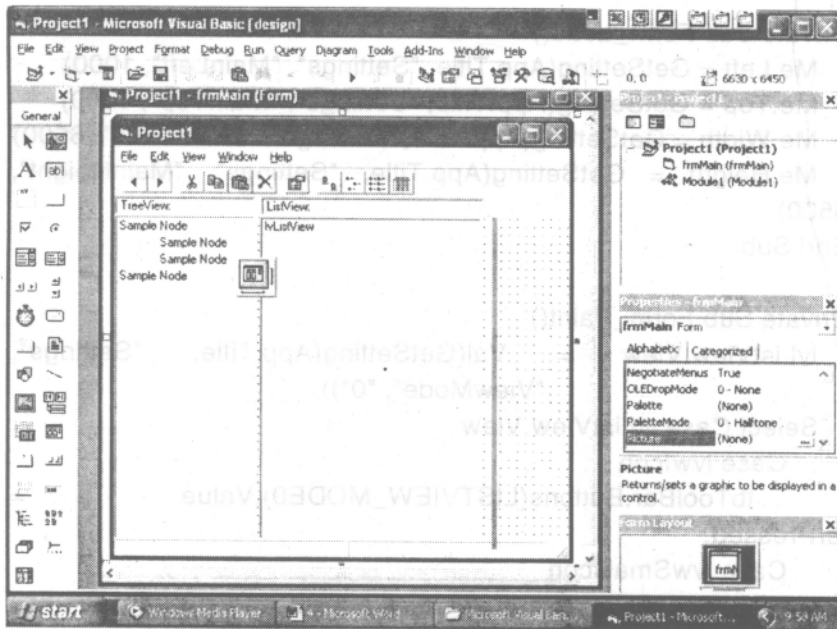
1. Khởi hoạt chương trình.
2. Trong khung cửa sổ biểu tượng New Project, kích biểu tượng VB Application Wizard, kích Open để mở khung hướng dẫn Application Wizard tạo khung cửa sổ chương trình.
3. Trong khung cửa sổ Application Wizard, nếu muốn dùng nền chương trình mới, kích ký hiệu ba chấm (...) nằm trong khung thông tin yêu cầu để mở khung thoại, nếu không, Next để chuyển sang khung màn hình hướng dẫn kế tiếp.



4. Trong khung màn hình hướng dẫn này, kích chọn thành phần Explore Style và đặt tên cho trình ứng dụng.
5. Sau khi đặt tên, nếu muốn thêm những thành phần khác kết hợp chung với màn hình chính Explore, kích Next và theo các hướng dẫn từ những khung hướng dẫn Wizard. Trong bài tập này, kích Finish để kết thúc.



6. Kích OK và màn hình Explorer có dạng như sau:



Dựa theo khung màn hình vừa thiết kế, bạn có thể kích đúp vào từng đối tượng và trong khung cửa sổ Properties, thay đổi các thuộc tính mặc định của chương trình theo ý muốn. Sau đó, mở khung cửa sổ Code để thay đổi mã lệnh cho phù hợp.

```
Const NAME_COLUMN = 0
Const TYPE_COLUMN = 1
Const SIZE_COLUMN = 2
Const DATE_COLUMN = 3
Private Declare Function OSWinHelp% Lib "user32" Alias
"WinHelpA" (ByVal hwnd&, ByVal HelpFile$, ByVal wCommand%,
dwData As Any)
```

```
Dim mbMoving As Boolean
```

Const sglSplitLimit = 500

Private Sub Form_Load()

Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)

Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)

Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)

Me.Height = GetSetting(App.Title, "Settings", "MainHeight",
6500)

End Sub

Private Sub Form_Paint()

lvListView.View = Val(GetSetting(App.Title, "Settings",
"ViewMode", "0"))

Select Case lvListView.View

Case lvwIcon

tbToolBar.Buttons(LISTVIEW_MODE0).Value =
tbrPressed

Case lvwSmallIcon

tbToolBar.Buttons(LISTVIEW_MODE1).Value =
tbrPressed

Case lvwList

tbToolBar.Buttons(LISTVIEW_MODE2).Value =
tbrPressed

Case lvwReport

tbToolBar.Buttons(LISTVIEW_MODE3).Value =
tbrPressed

End Select

End Sub

Private Sub Form_Unload(Cancel As Integer)

Dim i As Integer

'close all sub forms

For i = Forms.Count - 1 To 1 Step -1

Unload Forms(i)


```

Next
If Me.WindowState <> vbMinimized Then
    SaveSetting App.Title, "Settings", "MainLeft", Me.Left
    SaveSetting App.Title, "Settings", "MainTop", Me.Top
    SaveSetting App.Title, "Settings", "MainWidth", Me.Width
    SaveSetting App.Title, "Settings", "MainHeight", Me.Height
End If
SaveSetting App.Title, "Settings", "ViewMode", lvListView.View
End Sub

Private Sub Form_Resize()
    On Error Resume Next
    If Me.Width < 3000 Then Me.Width = 3000
    SizeControls imgSplitter.Left
End Sub

Private Sub imgSplitter_MouseDown(Button As Integer, Shift As
                                   Integer, X As Single, Y As
                                   Single)
    With imgSplitter
        picSplitter.Move .Left, .Top, .Width \ 2, .Height - 20
    End With
    picSplitter.Visible = True
    mbMoving = True
End Sub

Private Sub imgSplitter_MouseMove(Button As Integer, Shift As
                                   Integer, X As Single, Y As
                                   Single)
    Dim sglPos As Single

    If mbMoving Then
        sglPos = X + imgSplitter.Left
        If sglPos < sglSplitLimit Then
            picSplitter.Left = sglSplitLimit
        End If
    End If
End Sub

```

```

ElseIf sglPos > Me.Width - sglSplitLimit Then
    picSplitter.Left = Me.Width - sglSplitLimit
Else
    picSplitter.Left = sglPos
End If
End If
End Sub

Private Sub imgSplitter_MouseUp(Button As Integer, Shift As
                                Integer, X As Single, Y As Single)
    SizeControls picSplitter.Left
    picSplitter.Visible = False
    mbMoving = False
End Sub

Private Sub TreeView1_DragDrop(Source As Control, X As
                                Single, Y As Single)
    If Source = imgSplitter Then
        SizeControls X
    End If
End Sub

Sub SizeControls(X As Single)
    On Error Resume Next

    'set the width
    If X < 1500 Then X = 1500
    If X > (Me.Width - 1500) Then X = Me.Width - 1500
    tvTreeView.Width = X
    imgSplitter.Left = X
    lvListView.Left = X + 40
    lvListView.Width = Me.Width - (tvTreeView.Width + 140)
    lblTitle(0).Width = tvTreeView.Width
    lblTitle(1).Left = lvListView.Left + 20
    lblTitle(1).Width = lvListView.Width - 40

```

```

'set the top

If tbToolBar.Visible Then
    tvTreeView.Top = tbToolBar.Height + picTitles.Height
Else
    tvTreeView.Top = picTitles.Height
End If

lvListView.Top = tvTreeView.Top

'set the height
If sbStatusBar.Visible Then
    tvTreeView.Height = Me.ScaleHeight - (picTitles.Top +
                                         picTitles.Height +
                                         sbStatusBar.Height)
Else
    tvTreeView.Height = Me.ScaleHeight - (picTitles.Top +
                                         picTitles.Height)
End If

lvListView.Height = tvTreeView.Height
imgSplitter.Top = tvTreeView.Top
imgSplitter.Height = tvTreeView.Height
End Sub
Private Sub tbToolBar_ButtonClick(ByVal Button As
                                         MSCoMctlLib.Button)
On Error Resume Next
Select Case Button.Key
    Case "Back"
        'ToDo: Add 'Back' button code.
        MsgBox "Add 'Back' button code."
    Case "Forward"
        'ToDo: Add 'Forward' button code.
        MsgBox "Add 'Forward' button code."

```

```

Case "Cut"
    mnuEditCut_Click
Case "Copy"
    mnuEditCopy_Click
Case "Paste"
    mnuEditPaste_Click
Case "Delete"
    mnuFileDelete_Click
Case "Properties"
    mnuFileProperties_Click
Case "View Large Icons"
    lvListView.View = lvwIcon
Case "View Small Icons"
    lvListView.View = lvwSmallIcon
Case "View List"
    lvListView.View = lvwList
Case "View Details"
    lvListView.View = lvwReport
End Select
End Sub

Private Sub mnuHelpAbout_Click()
    MsgBox "Version " & App.Major & "." & App.Minor & "." &
App.Revision
End Sub

Private Sub mnuHelpSearchForHelpOn_Click()
    Dim nRet As Integer

    'if there is no helpfile for this project display a message to the
user
    'you can set the HelpFile for your application in the
    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help

```

associated with this project.", vbInformation, Me.Caption

Else

On Error Resume Next

nRet = OSWinHelp(Me.hwnd, App.HelpFile, 261, 0)

If Err Then

MsgBox Err.Description

End If

End If

End Sub

Private Sub mnuHelpContents_Click()

Dim nRet As Integer

'If there is no helpfile for this project display a message to the user

'you can set the HelpFile for your application in the

'Project Properties dialog

If Len(App.HelpFile) = 0 Then

MsgBox "Unable to display Help Contents. There is no Help associated with this project.", vbInformation, Me.Caption

Else

On Error Resume Next

nRet = OSWinHelp(Me.hwnd, App.HelpFile, 3, 0)

If Err Then

MsgBox Err.Description

End If

End If

End Sub

Private Sub mnuWindowArrangelcons_Click()

'ToDo: Add 'mnuWindowArrangelcons_Click' code.

MsgBox "Add 'mnuWindowArrangelcons_Click' code."

End Sub

```

Private Sub mnuWindowTileVertical_Click()
    'ToDo: Add 'mnuWindowTileVertical_Click' code.
    MsgBox "Add 'mnuWindowTileVertical_Click' code."
End Sub

Private Sub mnuWindowTileHorizontal_Click()
    'ToDo: Add 'mnuWindowTileHorizontal_Click' code.
    MsgBox "Add 'mnuWindowTileHorizontal_Click' code."
End Sub

Private Sub mnuWindowCascade_Click()
    'ToDo: Add 'mnuWindowCascade_Click' code.
    MsgBox "Add 'mnuWindowCascade_Click' code."
End Sub

Private Sub mnuWindowNewWindow_Click()
    'ToDo: Add 'mnuWindowNewWindow_Click' code.
    MsgBox "Add 'mnuWindowNewWindow_Click' code."
End Sub

Private Sub mnuViewWebBrowser_Click()
    'ToDo: Add 'mnuViewWebBrowser_Click' code.
    MsgBox "Add 'mnuViewWebBrowser_Click' code."
End Sub

Private Sub mnuViewOptions_Click()
    'ToDo: Add 'mnuViewOptions_Click' code.
    MsgBox "Add 'mnuViewOptions_Click' code."
End Sub

Private Sub mnuViewRefresh_Click()
    'ToDo: Add 'mnuViewRefresh_Click' code.
    MsgBox "Add 'mnuViewRefresh_Click' code."
End Sub

```

```

Private Sub mnuVAIByDate_Click()
    'ToDo: Add 'mnuVAIByDate_Click' code.
    ' lvListView.SortKey = DATE_COLUMN
End Sub

Private Sub mnuVAIByName_Click()
    'ToDo: Add 'mnuVAIByName_Click' code.
    ' lvListView.SortKey = NAME_COLUMN
End Sub

Private Sub mnuVAIBySize_Click()
    'ToDo: Add 'mnuVAIBySize_Click' code.
    ' lvListView.SortKey = SIZE_COLUMN
End Sub

Private Sub mnuVAIByType_Click()
    'ToDo: Add 'mnuVAIByType_Click' code.
    ' lvListView.SortKey = TYPE_COLUMN
End Sub

Private Sub mnuViewStatusBar_Click()
    mnuViewStatusBar.Checked = Not
mnuViewStatusBar.Checked
    sbStatusBar.Visible = mnuViewStatusBar.Checked
    SizeControls imgSplitter.Left
End Sub

Private Sub mnuViewToolBar_Click()
    mnuViewToolBar.Checked = Not mnuViewToolBar.Checked
    tbToolBar.Visible = mnuViewToolBar.Checked
    SizeControls imgSplitter.Left
End Sub

Private Sub mnuEditInvertSelection_Click()
    'ToDo: Add 'mnuEditInvertSelection_Click' code.

```

```

    MsgBox "Add 'mnuEditInvertSelection_Click' code."
End Sub

Private Sub mnuEditSelectAll_Click()
    'ToDo: Add 'mnuEditSelectAll_Click' code.
    MsgBox "Add 'mnuEditSelectAll_Click' code."
End Sub

Private Sub mnuEditPasteSpecial_Click()
    'ToDo: Add 'mnuEditPasteSpecial_Click' code.
    MsgBox "Add 'mnuEditPasteSpecial_Click' code."
End Sub

Private Sub mnuEditPaste_Click()
    'ToDo: Add 'mnuEditPaste_Click' code.
    MsgBox "Add 'mnuEditPaste_Click' code."
End Sub

Private Sub mnuEditCopy_Click()
    'ToDo: Add 'mnuEditCopy_Click' code.
    MsgBox "Add 'mnuEditCopy_Click' code."
End Sub

Private Sub mnuEditCut_Click()
    'ToDo: Add 'mnuEditCut_Click' code.
    MsgBox "Add 'mnuEditCut_Click' code."
End Sub

Private Sub mnuEditUndo_Click()
    'ToDo: Add 'mnuEditUndo_Click' code.
    MsgBox "Add 'mnuEditUndo_Click' code."
End Sub

Private Sub mnuFileClose_Click()
    'unload the form

```


Unload Me

End Sub

```
Private Sub mnuFileProperties_Click()  
    'ToDo: Add 'mnuFileProperties_Click' code.  
    MsgBox "Add 'mnuFileProperties_Click' code."  
End Sub
```

```
Private Sub mnuFileRename_Click()  
    'ToDo: Add 'mnuFileRename_Click' code.  
    MsgBox "Add 'mnuFileRename_Click' code."  
End Sub
```

```
Private Sub mnuFileDelete_Click()  
    'ToDo: Add 'mnuFileDelete_Click' code.  
    MsgBox "Add 'mnuFileDelete_Click' code."  
End Sub
```

```
Private Sub mnuFileNew_Click()  
    'ToDo: Add 'mnuFileNew_Click' code.  
    MsgBox "Add 'mnuFileNew_Click' code."  
End Sub
```

```
Private Sub mnuFileSendTo_Click()  
    'ToDo: Add 'mnuFileSendTo_Click' code.  
    MsgBox "Add 'mnuFileSendTo_Click' code."  
End Sub
```

```
Private Sub mnuFileFind_Click()  
    'ToDo: Add 'mnuFileFind_Click' code.  
    MsgBox "Add 'mnuFileFind_Click' code."  
End Sub
```

```
Private Sub mnuFileOpen_Click()
```

Dim sFile As String

With dlgCommonDialog

.DialogTitle = "Open"

.CancelError = False

'ToDo: set the flags and attributes of the common dialog control

.Filter = "All Files (*.*)|*.*"

.ShowOpen

If Len(.FileName) = 0 Then

Exit Sub

End If

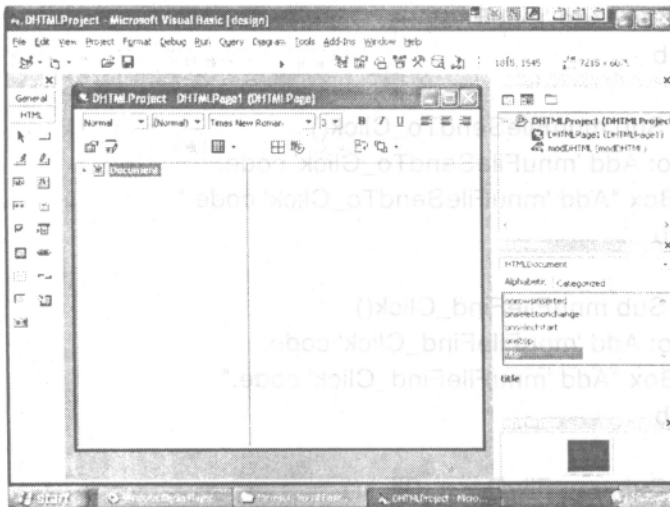
sFile = .FileName

End With

'ToDo: add code to process the opened file

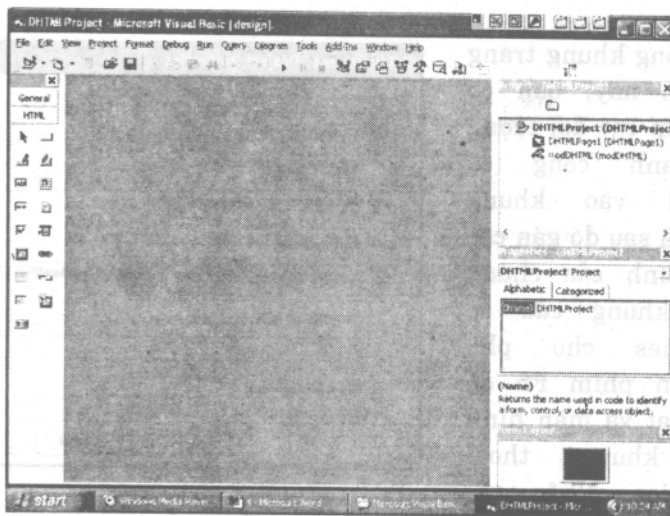
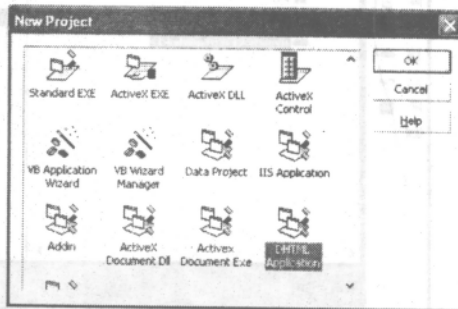
End Sub

Bài 20: Thiết kế trang DHTML

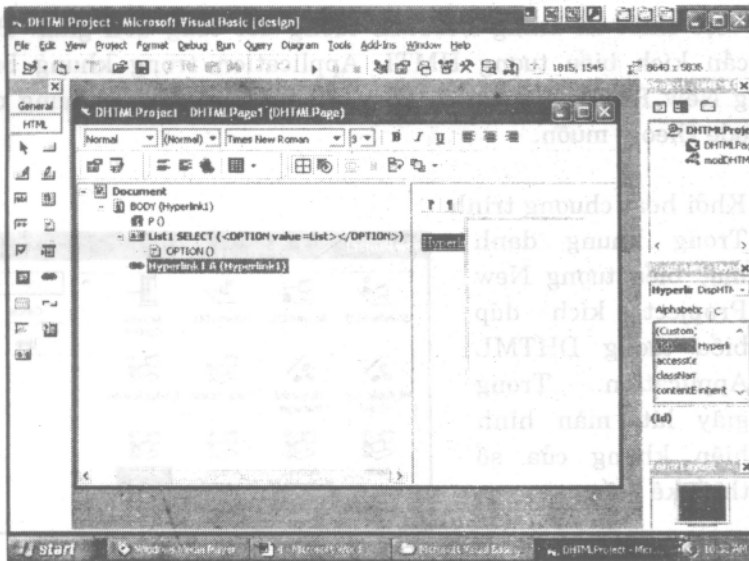


Việc thiết kế trang DHTML tương đối cũng đơn giản. Bạn chỉ cần kích biểu tượng HMTL Application trong khung biểu tượng New Project và sau đó thay đổi mã lệnh trong khung cửa sổ Code theo ý muốn.

1. Khởi hoạt chương trình.
2. Trong khung danh mục biểu tượng New Propject, kích đúp biểu tượng DHTML Application. Trong giây lát, màn hình hiện khung cửa sổ thiết kế trống.

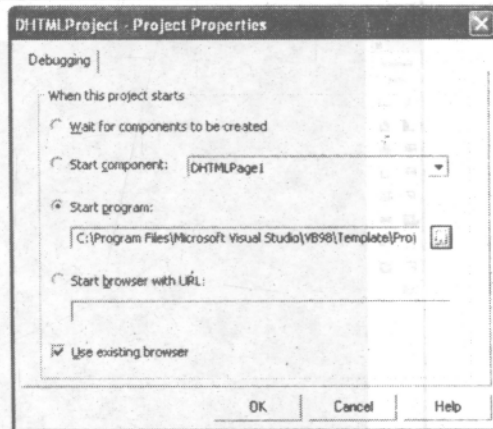


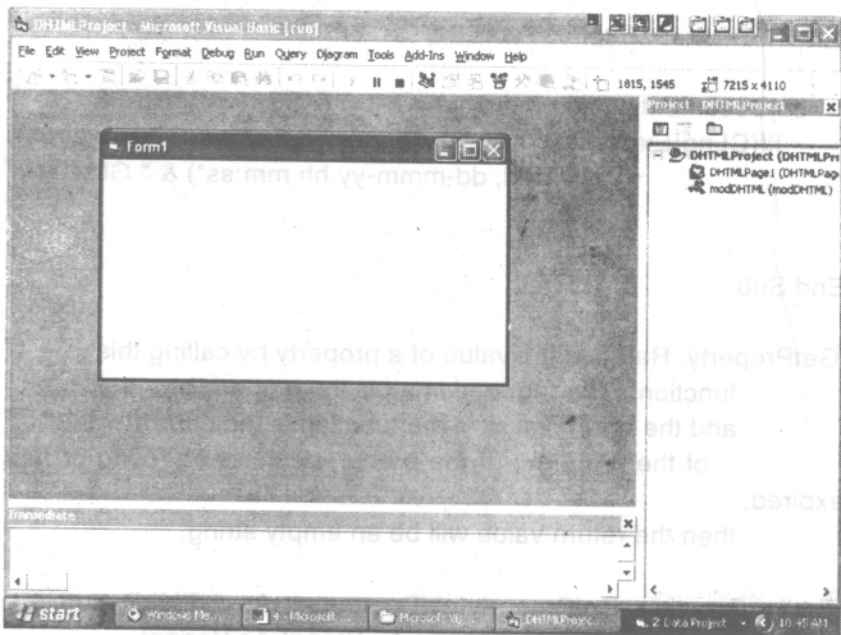
Lưu ý là trong khung cửa sổ Project Manager nằm bên phải khung cửa sổ thiết kế có chứa hai đối tượng của trang DHTML. Một thể hiện trang DHTML và một hiện khung cửa sổ mã lệnh. Kích công cụ DHTMLPage1 để hiện khung cửa sổ trang DHTML.



Trong khung trang DHTML này, bạn có thể thêm các đối tượng từ thanh công cụ General vào khung trang và sau đó gán các thuộc tính của chúng trong khung của sổ Properties cho phù hợp. Ấn phím F5 để điều hoạt và màn hình hiện khung thoại Properties. Nhập các thành phần cần thiết và kích OK.

Với thí dụ trong khung thoại trên, sau khi kích OK, màn hình thiết kế có dạng như sau:





Kích nút lệnh Stop để trở về khung màn hình thiết kế. Sau đó, trong khung cửa sổ Project Manager, kích đúp công cụ ModDHTML để hiện khung mã lệnh Code và từ đây, bạn có thể thay đổi hoặc thêm các hàm lệnh của các đối tượng cho phù hợp.

PutProperty: Store information in a cookie by calling this function.

The required inputs are the named Property and the value of the property you would like to store.

Optional inputs are:

expires : specifies a date that defines the valid life time of the property. Once the expiration date has been reached, the property will no longer be stored or

given out.

```
Public Sub PutProperty(objDocument As HTMLDocument,
    strName As String, vntValue As Variant, Optional
```

Expires As Date)

```
objDocument.cookie = strName & "=" & CStr(vntValue) & _  
    If(CLng(Expires) = 0, "", "; expires=" & Format(CStr(Expires),  
        "ddd, dd-mmm-yy hh:mm:ss") & " GMT") ' &
```

End Sub

'GetProperty: Retrieve the value of a property by calling this
' function. The required input is the named Property,
' and the return value of the function is the current value
' of the property. If the proeprty cannot be found or has
expired,
' then the return value will be an empty string.
'

Public Function GetProperty(objDocument As HTMLDocument,
 strName As String) As Variant

Dim aryCookies() As String

Dim strCookie As Variant

On Local Error GoTo NextCookie

'Split the document cookie object into an array of cookies.

aryCookies = Split(objDocument.cookie, ";")

For Each strCookie In aryCookies

If Trim(VBA.Left(strCookie, InStr(strCookie, "=") - 1)) =
 Trim(strName) Then

GetProperty = Trim(Mid(strCookie, InStr(strCookie, "=") +
1))

Exit Function

End If

NextCookie:

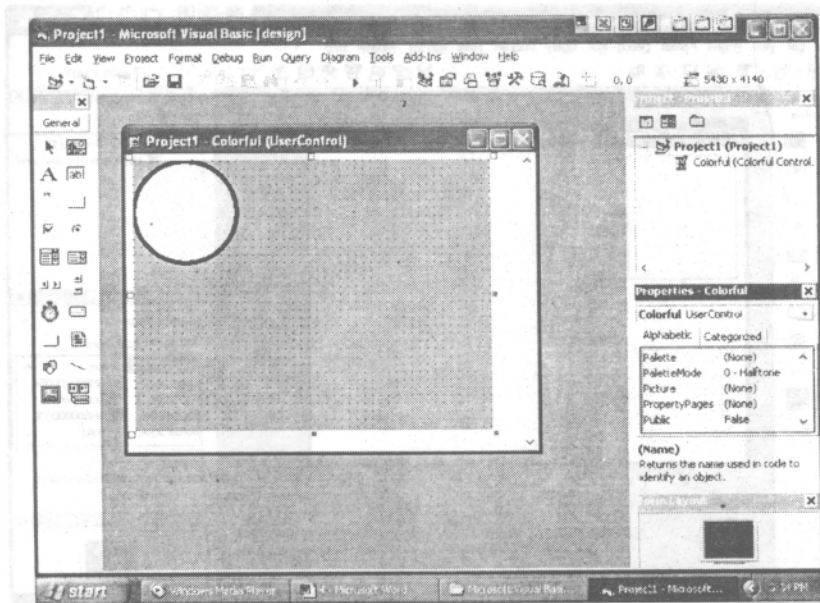
Err = 0

Next strCookie

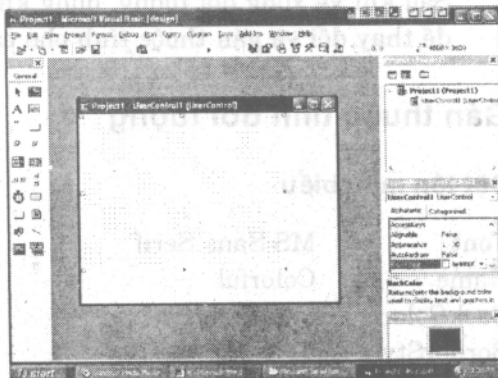
End Function

Bài 21: Vẽ đối tượng

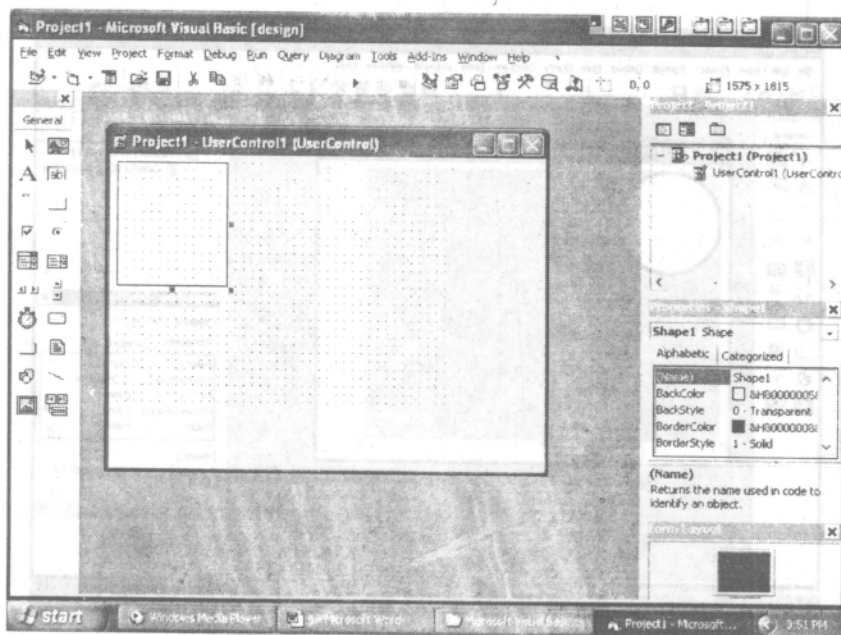
Trong bài tập này, bạn vẽ một đối tượng bất kỳ trong màn hình thiết kế để làm quen và sau đó có thể áp dụng thủ tục này để vẽ những hình ảnh phức tạp khác.



1. Khởi hoạt chương trình.
2. Trong khung danh mục biểu tượng New Propject, kích đúp biểu tượng Active Control. Trong giây lát, màn hình hiện khung của sổ thiết kế trống.



3. Để bắt đầu vẽ đối tượng, kích công cụ Shape trong thanh công cụ General. Chọn vị trí trong khung màn hình thiết kế kéo, ấn chìm nút mouse và kéo để tạo dạng. Thả nút mouse sau khi đối tượng có hình dạng theo ý muốn. Nếu cần thiết, dùng chức năng kích và kéo để định lại vị trí.



4. Sau khi vẽ xong đối tượng, dùng khung thuộc tính Properties để thay đổi và gán thuộc tính cho các đối tượng.

Gán thuộc tính đối tượng

Đặt tên mẫu biểu

Font	MS Sans Serif
Name	Colorful
BackColor	màu cam
BorderStyle	0 - None

Gán thuộc tính đối tượng

Name	Shape1
BorderColor	màu đen
BorderStyle	1 – Solid
BorderWidth	5
FillColor	màu vàng
Shape	2 - Oval

Sau khi thay đổi các thuộc tính cho đối tượng, mở khung của sổ Code và thay đổi mã lệnh chương trình theo ý muốn.

```
Public Property Get Color() As OLE_COLOR  
    .Color = Shape1.FillColor  
End Property
```

```
Public Property Let Color(ByVal c As OLE_COLOR)  
    Shape1.FillColor = c  
End Property
```

```
Private Sub UserControl_Click()  
    Shape1.FillColor = &HFF0000  
End Sub
```

```
Private Sub UserControl_EnterFocus()  
    Shape1.BorderColor = &HFFFFFF  
End Sub
```

```
Private Sub UserControl_ExitFocus()  
    Shape1.BorderColor = &H0  
End Sub
```

```
Private Sub UserControl_Resize()  
    d = Shape1.BorderWidth / 2  
    Shape1.Left = d  
    Shape1.Top = d
```

```

Shape1.Width = ScaleWidth - d * 2
Shape1.Height = ScaleHeight - d * 2
End Sub

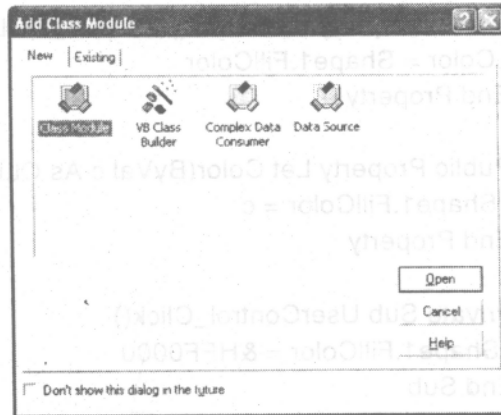
```

Bài 22: Thêm một Class vào mẫu biểu

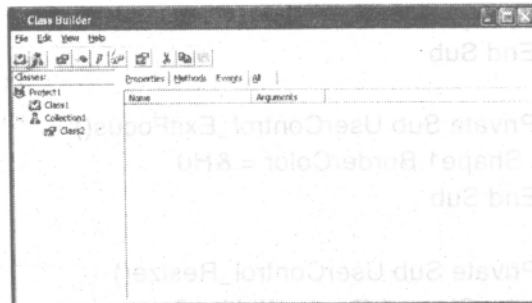
Bạn có thể thêm một khối Class vào mẫu biểu hiện hành rồi sau đó thay đổi mã chương trình trong khung cửa sổ Code.

1. Mở lại tập tin mẫu biểu muốn thêm khối Class.

2. Từ menu Project, chọn Add Class Module. Màn hình hiện khung biểu tượng Add Class Module trên màn hình.



3. Kích đúp biểu tượng VB Class Builder, khung cửa sổ VB Builder hiện lên màn hình. Từ đây, bạn có thể bắt đầu thiết lập các khối Class liên quan cho mẫu biểu.



4. Sau khi kết thúc quy trình thêm các khối Class vào mẫu biểu, để quan sát và thay đổi mã chương trình, thoát khỏi khung cửa sổ Class Builder. Kích lại menu Project, chọn Add Class Module và trong khung cửa sổ biểu tượng Add Class Module, kích đúp biểu tượng Complex Data Consumer. Màn hình hiện khung cửa sổ Code với chương trình đã lập sẵn. Thêm hoặc thay đổi các hàm lệnh cho các đối tượng vừa thiết kế.

Option Explicit

' TODO: declare a recordset variable to be the datasource for this class. For example:

'Private WithEvents RS As ADODB.RecordSet

Public Property Get DataSource() As DataSource

' TODO: return the datasource object for this class.

'For example:

'Set DataSource = RS.DataSource

End Property

Public Property Set DataSource(ByVal objDataSource As DataSource)

' TODO: set the datasource object for this class. For example:

'Set RS.DataSource = objDataSource

'PropertyChanged "DataSource"

End Property

Public Property Get DataMember() As DataMember

' TODO: return the DataMember. For example:

'Set DataMember = RS.DataMember

End Property

Public Property Let DataMember(ByVal DataMember As DataMember)

' TODO: assign a new DataMember. For example:

'Set RS.DataMember = DataMember

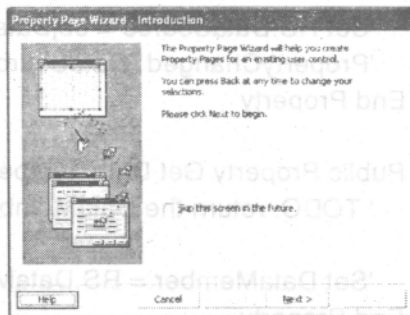
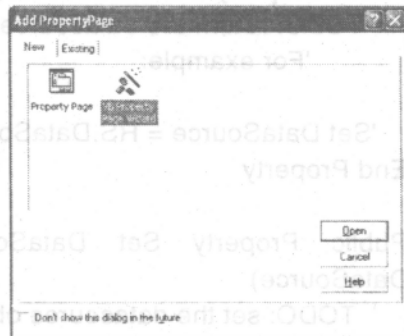
'PropertyChanged "DataMember"

End Property

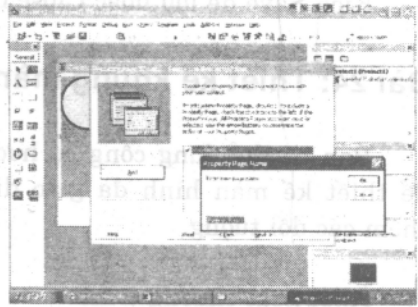
Bài 23: Thêm trang thuộc tính vào mẫu biểu thiết kế

Sau khi thiết kế các đối tượng ActiveX Control, bạn có thể thêm các trang thuộc tính của chúng vào mẫu biểu bằng cách:

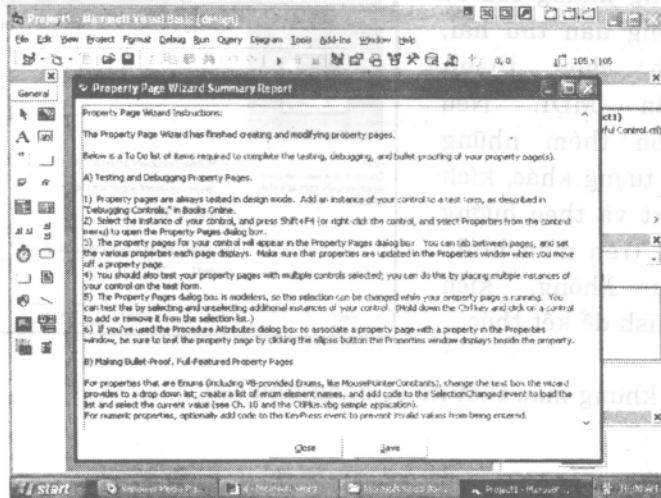
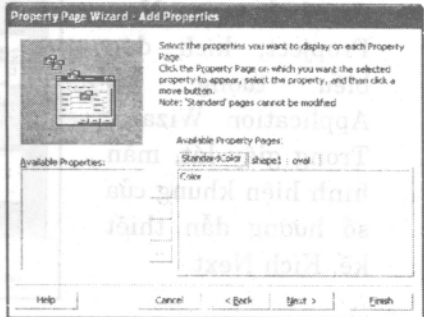
1. Mở lại mẫu biểu ActiveX Control lên màn hình. Trong thí dụ này, bạn mở lại tập tin mẫu biểu Colorful lên màn hình.
2. Kích menu Project và chọn Add Property Page từ menu xổ. Màn hình hiện khung biểu tượng Add Property Page. Kích chọn thành phần muốn lập trang thuộc tính.
3. Muốn lập trang thuộc tính cho từng thành phần của đối tượng trong mẫu biểu, kích đúp biểu tượng VB Property Page Wizard. Màn hình hiện khung hướng dẫn Wizard. Kích Next.



4. Trong khung hướng dẫn này, muốn thêm thành phần khác trong mẫu biểu vào trang thuộc tính, kích thước lệnh Add và đặt tên cho thành phần trong khung thoại. Sau đó, kích OK để trở về khung hướng dẫn. Kích Next để chuyển sang khung màn hình hướng dẫn kế tiếp.



5. Trong khung màn hình hướng dẫn này, kích từng thuộc lệnh để quan sát tên thành phần muốn thêm trang thuộc tính. Kích Finish và trang thuộc tính có dạng như sau:



Kích Save để lưu hoặc Close để thoát và không lưu lại.

Bài 24: Thiết kế khung màn hình đa giao diện

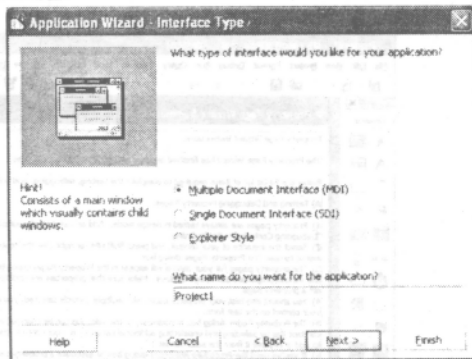
Bạn có thể dùng công cụ hướng dẫn VB Application Wizard để thiết kế màn hình đa giao diện và sau đó tự thay đổi hoặc thêm các đối tượng.

1. Khởi hoạt chương trình.

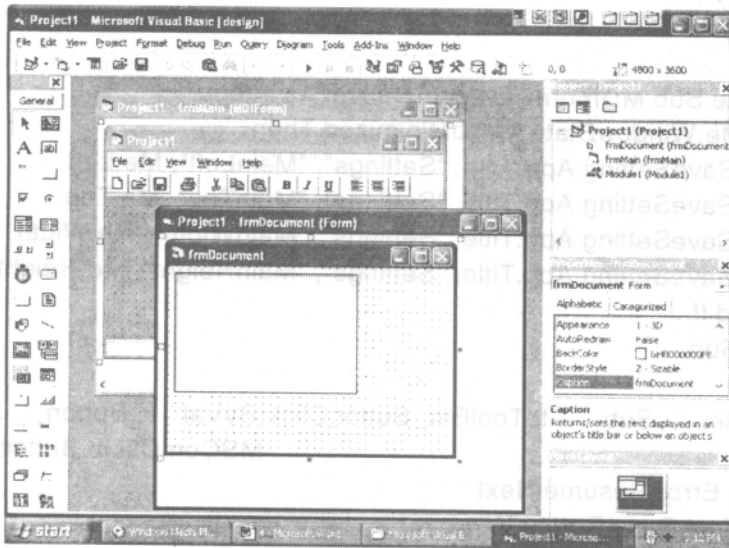
2. Trong khung danh mục biểu tượng New Project, kích đúp biểu tượng VB Application Wizard. Trong giây lát, màn hình hiện khung cửa sổ hướng dẫn thiết kế. Kích Next.



3. Trong khung cửa sổ hướng dẫn thứ hai, kích chọn thành phần MDI. Nếu muốn thêm những đối tượng khác, kích Next và theo hướng dẫn trên màn hình. Nếu không, kích Finish để kết thúc.



Và khung mẫu thiết kế có dạng như sau:



Mở khung cửa sổ Code và thay đổi mã chương trình cho phù hợp.

```
Private Sub MDIForm_Load()
    Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
    Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
    Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
    Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
```

```
    LoadNewDoc
End Sub
```

```
Private Sub LoadNewDoc()
    Static IDocumentCount As Long
    Dim frmD As frmDocument
    IDocumentCount = IDocumentCount + 1
    Set frmD = New frmDocument
    frmD.Caption = "Document " & IDocumentCount
    frmD.Show
```



```

        ActiveForm.rtfText.SelItalic = Not
            ActiveForm.rtfText.SelItalic
        Button.Value = If(ActiveForm.rtfText.SelItalic, tbrPressed,
            tbrUnpressed)
    Case "Underline"
        ActiveForm.rtfText.SelUnderline = Not
            ActiveForm.rtfText.SelUnderline
        Button.Value = If(ActiveForm.rtfText.SelUnderline,
            tbrPressed, tbrUnpressed)
    Case "Align Left"
        ActiveForm.rtfText.SelAlignment = rtfLeft
    Case "Center"
        ActiveForm.rtfText.SelAlignment = rtfCenter
    Case "Align Right"
        ActiveForm.rtfText.SelAlignment = rtfRight
    End Select
End Sub

Private Sub mnuHelpAbout_Click()
    MsgBox "Version " & App.Major & "." & App.Minor & "." &
        App.Revision
End Sub

Private Sub mnuHelpSearchForHelpOn_Click()
    Dim nRet As Integer

    'if there is no helpfile for this project display a message to the
    user you can set the HelpFile for your application in the
    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help
        associated with this project.", vbInformation, Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 261, 0)
    End If
End Sub

```

```

        If Err Then
            MsgBox Err.Description
        End If
    End If

End Sub

Private Sub mnuHelpContents_Click()
    Dim nRet As Integer

    'if there is no helpfile for this project display a message to the
    user you can set the HelpFile for your application in the
    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help
        associated with this project.", vbInformation, Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 3, 0)
        If Err Then
            MsgBox Err.Description
        End If
    End If

End Sub

End Sub

Private Sub mnuWindowArrangelcons_Click()
    Me.Arrange vbArrangelcons
End Sub

Private Sub mnuWindowTileVertical_Click()
    Me.Arrange vbTileVertical
End Sub

```

```
Private Sub mnuWindowTileHorizontal_Click()
    Me.Arrange vbTileHorizontal
End Sub
```

```
Private Sub mnuWindowCascade_Click()
    Me.Arrange vbCascade
End Sub
```

```
Private Sub mnuWindowNewWindow_Click()
    LoadNewDoc
End Sub
```

```
Private Sub mnuViewWebBrowser_Click()
    'ToDo: Add 'mnuViewWebBrowser_Click' code.
    MsgBox "Add 'mnuViewWebBrowser_Click' code."
End Sub
```

```
Private Sub mnuViewOptions_Click()
    'ToDo: Add 'mnuViewOptions_Click' code.
    MsgBox "Add 'mnuViewOptions_Click' code."
End Sub
```

```
Private Sub mnuViewRefresh_Click()
    'ToDo: Add 'mnuViewRefresh_Click' code.
    MsgBox "Add 'mnuViewRefresh_Click' code."
End Sub
```

```
Private Sub mnuViewStatusBar_Click()
    mnuViewStatusBar.Checked = Not
        mnuViewStatusBar.Checked
    sbStatusBar.Visible = mnuViewStatusBar.Checked
End Sub
```

```
Private Sub mnuViewToolbar_Click()
    mnuViewToolbar.Checked = Not mnuViewToolbar.Checked
```

```

    tbToolBar.Visible = mnuViewToolBar.Checked
End Sub

Private Sub mnuEditPasteSpecial_Click()
    'ToDo: Add 'mnuEditPasteSpecial_Click' code.
    MsgBox "Add 'mnuEditPasteSpecial_Click' code."
End Sub

Private Sub mnuEditPaste_Click()
    On Error Resume Next
    ActiveForm.rtfText.SelRTF = Clipboard.GetText

End Sub

Private Sub mnuEditCopy_Click()
    On Error Resume Next
    Clipboard.SetText ActiveForm.rtfText.SelRTF

End Sub

Private Sub mnuEditCut_Click()
    On Error Resume Next
    Clipboard.SetText ActiveForm.rtfText.SelRTF
    ActiveForm.rtfText.SelText = vbNullString

End Sub

Private Sub mnuEditUndo_Click()
    'ToDo: Add 'mnuEditUndo_Click' code.
    MsgBox "Add 'mnuEditUndo_Click' code."
End Sub

Private Sub mnuFileExit_Click()
    'unload the form
    Unload Me

```

End Sub

```
Private Sub mnuFileSend_Click()  
    'ToDo: Add 'mnuFileSend_Click' code.  
    MsgBox "Add 'mnuFileSend_Click' code."  
End Sub
```

```
Private Sub mnuFilePrint_Click()  
    On Error Resume Next  
    If ActiveForm Is Nothing Then Exit Sub
```

```
    With dlgCommonDialog  
        .DialogTitle = "Print"  
        .CancelError = True  
        .Flags = cdIPDReturnDC + cdIPDNoPageNums  
        If ActiveForm.rtfText.SelLength = 0 Then  
            .Flags = .Flags + cdIPDAllPages  
        Else  
            .Flags = .Flags + cdIPDSelection  
        End If  
        .ShowPrinter  
        If Err <> MSComDlg.cdICancel Then  
            ActiveForm.rtfText.SelPrint .hDC  
        End If  
    End With
```

End Sub

```
Private Sub mnuFilePrintPreview_Click()  
    'ToDo: Add 'mnuFilePrintPreview_Click' code.  
    MsgBox "Add 'mnuFilePrintPreview_Click' code."  
End Sub
```

```
Private Sub mnuFilePageSetup_Click()  
    On Error Resume Next
```

```

    With dlgCommonDialog
        .DialogTitle = "Page Setup"
        .CancelError = True
        .ShowPrinter
    End With

End Sub

Private Sub mnuFileProperties_Click()
    'ToDo: Add 'mnuFileProperties_Click' code.
    MsgBox "Add 'mnuFileProperties_Click' code."
End Sub

Private Sub mnuFileSaveAll_Click()
    'ToDo: Add 'mnuFileSaveAll_Click' code.
    MsgBox "Add 'mnuFileSaveAll_Click' code."
End Sub

Private Sub mnuFileSaveAs_Click()
    Dim sFile As String

    If ActiveForm Is Nothing Then Exit Sub

    With dlgCommonDialog
        .DialogTitle = "Save As"
        .CancelError = False
        'ToDo: set the flags and attributes of the common dialog
            control
        .Filter = "All Files (*.*)|*.*"
        .ShowSave
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        sFile = .FileName
    End With

```

```

ActiveForm.Caption = sFile
ActiveForm.rtfText.SaveFile sFile

End Sub

Private Sub mnuFileSave_Click()
    Dim sFile As String
    If Left$(ActiveForm.Caption, 8) = "Document" Then
        With dlgCommonDialog
            .DialogTitle = "Save"
            .CancelError = False
            'ToDo: set the flags and attributes of the common dialog
                control
            .Filter = "All Files (*.*)|*.*"
            .ShowSave
            If Len(.FileName) = 0 Then
                Exit Sub
            End If
            sFile = .FileName
        End With
        ActiveForm.rtfText.SaveFile sFile
    Else
        sFile = ActiveForm.Caption
        ActiveForm.rtfText.SaveFile sFile
    End If
End Sub

End Sub

Private Sub mnuFileClose_Click()
    'ToDo: Add 'mnuFileClose_Click' code.
    MsgBox "Add 'mnuFileClose_Click' code."
End Sub

Private Sub mnuFileOpen_Click()
    Dim sFile As String

```

If ActiveForm Is Nothing Then LoadNewDoc

With dlgCommonDialog

.DialogTitle = "Open"

.CancelError = False

ToDo: set the flags and attributes of the common dialog control

.Filter = "All Files (*.*)|*.*"

.ShowOpen

If Len(.FileName) = 0 Then

Exit Sub

End If

sFile = .FileName

End With

ActiveForm.rtfText.LoadFile sFile

ActiveForm.Caption = sFile

End Sub

Private Sub mnuFileNew_Click()

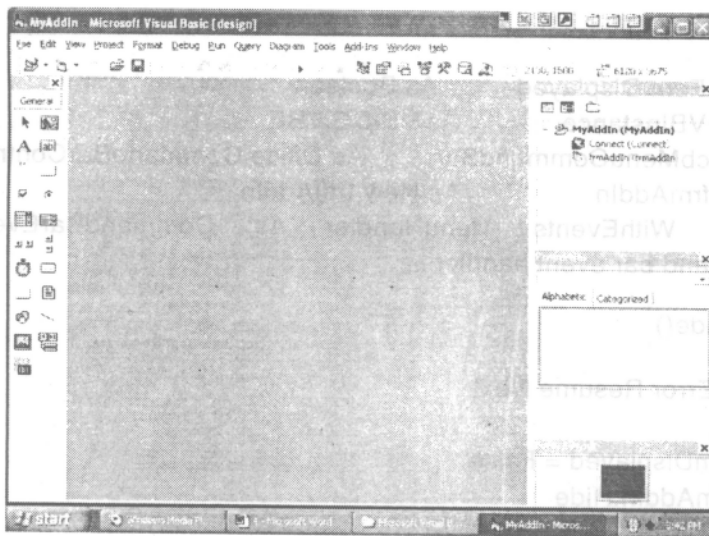
LoadNewDoc

End Sub

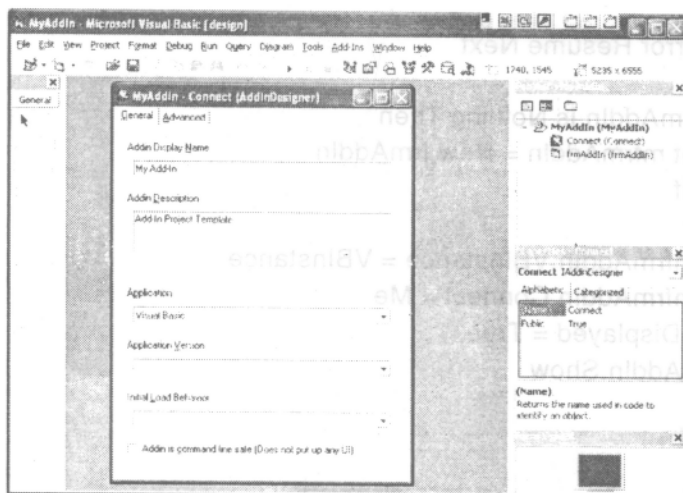
Bài 25: Thiết kế khung màn hình Add-Ins

1. Khởi hoạt chương trình.
2. Kích đúp biểu tượng Addin trong khung cửa sổ biểu tượng New Project và màn hình hiện khung cửa sổ thiết kế trống với hai đối tượng trong khung cửa sổ Project Explorer.





3. Kích đúp biểu tượng Connect trong khung cửa sổ Project Manager để quan sát đối tượng.



4. Ấn phím F5 và kích đúp biểu tượng Connect để mở khung cửa sổ Code và nếu cần thiết, thay đổi mã lệnh cho phù hợp.

Option Explicit

```
Public FormDisplayed      As Boolean
Public VBInstance        As VBIDE.VBE
Dim mcbMenuCommandBar    As Office.CommandBarControl
Dim mfrmAddIn            As New frmAddIn
Public WithEvents MenuHandler As CommandBarEvents
'command bar event handler
```

```
Sub Hide()
```

```
    On Error Resume Next
```

```
    FormDisplayed = False
    mfrmAddIn.Hide
```

```
End Sub
```

```
Sub Show()
```

```
    On Error Resume Next
```

```
    If mfrmAddIn Is Nothing Then
        Set mfrmAddIn = New frmAddIn
    End If
```

```
    Set mfrmAddIn.VBInstance = VBInstance
    Set mfrmAddIn.Connect = Me
    FormDisplayed = True
    mfrmAddIn.Show
```

```
End Sub
```

```
'-----
'this method adds the Add-In to VB
'-----
```

```

Private Sub AddInInstance_OnConnection(ByVal Application As
Object, ByVal ConnectMode As
AddInDesignerObjects.ext_ConnectMode, ByVal AddInInst As
Object, custom() As Variant)
    On Error GoTo error_handler

    'save the vb instance
    Set VBInstance = Application

    'this is a good place to set a breakpoint and
    'test various addin objects, properties and methods
    Debug.Print VBInstance.FullName

    If ConnectMode = ext_cm_External Then
        'Used by the wizard toolbar to start this wizard
        Me.Show
    Else
        Set mcbMenuCommandBar = AddToAddInCommandBar("My
            AddIn")

        'sink the event
        Set Me.MenuHandler =
VBInstance.Events.CommandBarEvents(mcbMenuCommandBar)
    End If

    If ConnectMode = ext_cm_AfterStartup Then
        If GetSetting(App.Title, "Settings", "DisplayOnConnect", "0")
            = "1" Then
            'set this to display the form on connect
            Me.Show
        End If
    End If

    Exit Sub

error_handler:

```

MsgBox Err.Description

End Sub

```
-----  
'this method removes the Add-In from VB  
-----
```

```
Private Sub AddinInstance_OnDisconnection(ByVal RemoveMode  
As AddInDesignerObjects.ext_DisconnectMode, custom() As  
Variant)
```

```
    On Error Resume Next
```

```
    'delete the command bar entry  
    mcbMenuCommandBar.Delete
```

```
    'shut down the Add-In
```

```
    If FormDisplayed Then
```

```
        SaveSetting App.Title, "Settings", "DisplayOnConnect", "1"
```

```
        FormDisplayed = False
```

```
    Else
```

```
        SaveSetting App.Title, "Settings", "DisplayOnConnect", "0"
```

```
    End If
```

```
    Unload mfrmAddIn
```

```
    Set mfrmAddIn = Nothing
```

End Sub

```
Private Sub IDTextensibility_OnStartupComplete(custom() As  
Variant)
```

```
    If GetSetting(App.Title, "Settings", "DisplayOnConnect", "0") =  
        "1" Then
```

```
        'set this to display the form on connect
```

```
        Me.Show
```

```
    End If
```

End Sub

```

'this event fires when the menu is clicked in the IDE
Private Sub MenuHandler_Click(ByVal CommandBarControl As
Object, handled As Boolean, CancelDefault As Boolean)
    Me.Show
End Sub

Function AddToAddInCommandBar(sCaption As String) As
Office.CommandBarControl
    Dim cbMenuCommandBar As Office.CommandBarControl
        'command bar object
    Dim cbMenu As Object

    On Error GoTo AddToAddInCommandBarErr

    'see if we can find the Add-Ins menu
    Set cbMenu = VBInstance.CommandBars("Add-Ins")
    If cbMenu Is Nothing Then
        'not available so we fail
        Exit Function
    End If

    'add it to the command bar
    Set cbMenuCommandBar = cbMenu.Controls.Add(1)
    'set the caption
    cbMenuCommandBar.Caption = sCaption

    Set AddToAddInCommandBar = cbMenuCommandBar

    Exit Function

AddToAddInCommandBarErr:

End Function

```

NỘI DUNG

	Trang
1. Môi trường Visual Basic	7
Màn hình chương trình	8
Tóm lược các lệnh menu	10
Lập trình Visual Basic	29
Sử dụng cá câu lệnh	30
Khai báo biến	36
Từ khóa đối chiếu ME	40
Toán tử	40
Đặt tên đối tượng	41
2. Bài tập thực hành	42
Tạo các mẫu biểu (Form)	42
Tạo khung thoại Dialog Caption	42
Tạo khung thoại About	48
Tạo khung nhập mật mã Login	58
Tạo khung thoại ODBC Logon	63
Tạo tập tin Tip OF the Day	73
Tạo tập tin Giới thiệu sản phẩm	80
Tạo khung cửa sổ trình duyệt Web	85
Tạo các nút khiển (Control)	91
Tạo danh mục nút lệnh	91
Tạo khung chuyển danh sách	97
Tạo khung cửa sổ cây danh mục	105
Tạo khung Wizard	110
Tạo khung Wizard mẫu	118
Tạo khung thông báo	127
Tạo khung cài đặt chương trình	130
Tạo khung kết nhóm đối tượng	136
Tạo khung thoại Disk Space	147
Tạo khung đường dẫn	155
Tạo khung màn hình Copy	163
Tạo khung cửa sổ trình ứng dụng	166
Tạo khung màn hình Explorer	181
Thiết kế trang DHTML	194
Vẽ đối tượng	199
Thêm Class vào mẫu biểu	202
Thêm trang thuộc tính vào mẫu biểu	204
Thiết kế màn hình đa giao diện	206
Thiết kế khung màn hình Add-Ins	216

TỰ HỌC VISUAL BASIC & BÀI TẬP THỰC HÀNH

Ban biên dịch:

HOÀN VŨ

Chịu trách nhiệm xuất bản:

TS. QUÁCH THU NGUYỆT

Biên tập : **YÊN THẢO**
Vẽ bìa : **VŨ TÚ**
Sửa bản in : **QUỲNH LÂM**

Tổ chức cá nhân liên doanh:

VŨ HÙNG

Đ.T: 5.111431 - 0908.101171

NHÀ XUẤT BẢN TRẺ

161B Lý Chính Thắng - Q.3 - T.p. Hồ Chí Minh

ĐT: 9316289 - 9316211 - 8465595 - 8465596 - 9350973

Fax: 08.8437450

E-Mail: xnbtre@hcm.vnn.vn

Website: <http://www.nxbtre.com.vn>

CHI NHÁNH NHÀ XUẤT BẢN TRẺ TẠI HÀ NỘI

Số 20 ngõ 91 Nguyễn Chí Thanh - Quận Đống Đa - Hà Nội

ĐT: (04)7734544 - Fax: (04)7734544

E-mail: vanphongnxbtre@hn.vnn.vn

In 800 cuốn, khổ 14,5cm x 20,5cm, tại XN IN Tân Bình. Số
đăng ký KHXB 927/18/CXB do Cục Xuất bản cấp ngày 31/7/2003
và giấy trích ngang KHXB số 1119/KHXB/2004.

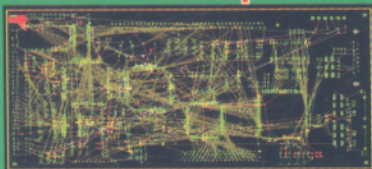
In xong và nộp lưu chiểu 10/2004.

HOÀNG VĂN ĐĂNG

Layout
.....
PSpice
.....
Capture
.....
Capture CIS
orcad
a Cadence product family
9.2

PHẦN MỀM

THIẾT KẾ MẠCH IN



HOÀNG VĂN ĐĂNG

OrCAD
PSPICE 9

PHẦN MỀM PHÂN TÍCH MẠCH



NHÀ XUẤT BẢN TRẺ

HOÀNG VĂN ĐĂNG

75
BÀI TẬP
PASCAL



HOÀNG VĂN ĐĂNG

OrCAD
CAPTURE CIS 9

PHẦN MỀM VẼ MẠCH NGUYÊN LÝ



NHÀ XUẤT BẢN TRẺ

Giá: 24.000Đ

tự học visual basic và bài tập



1 004091 400755
24.000 VNĐ