


The “House” Program

The “House” program uses labels to create “Hot Spots”. Labels cover various parts of the picture. When the user moves the mouse to one of the labels, the mouse pointer changes to a hand, and the tool tip pops up with a description of the part. This program requires a picture of a house. You can use the one provided with this lesson, or find one you like on the Internet:

- Go to www.google.com.
 - Click on Images.
 - Type house and press Enter.
 - Double click a picture you like to open it full size.
 - Right-click and save to disk.
1. Start a new windows application and name it House.
 2. In the property window select any image as the BackgroundImage. Notice that the image is tiled. We would like to use a picture without tiling. Because the user can resize the form, there is no way to keep the image from tiling. Instead of putting the image on the form we will use a panel.
 3. In the property window, delete the image by selecting the property and pressing delete.
 4. Add a Panel, PnlHouse and select the house picture for the Image. Drag the sides of the panel so that the picture of the house is completely visible.
 5. Add a tooltip  control to the form.
 6. From the toolbox select a label and drag it onto the panel. Make sure that it is contained in the panel by dragging the panel. The label should move with the panel. If the label is not in the panel, click the label and type “CTRL+X” to put it on the clipboard. Click the panel and type “CTRL+V” to paste it inside the panel.

An Invisible Label

Name the label LblDoor. Set Autosize to false and the text to nothing. Set the background to Transparent. (Transparent is at the top of the Web tab in the color dialog.) Drag LblDoor so that it is over the door in the picture.



Set the tooltip property to “Door”.

Run the program and move the mouse over the door. You should see the word “door” pop up.

Select ToolTip1 and look at the properties. Change ShowAlways to True. Adjust any of the variables that refer to Delay and run the program again. Adjust the values so that you know what they do and find settings you like.

Copy LblDoor and paste it inside the panel. Change the name to Lbl Window and the tooltip to "Window."

1. Add a second Tooltip. Leave the default settings for tooltip2.
2. Change the properties for LblDoor so that ToolTip for Tooltip1 is "Door", and "Old and brown" for the second one.
3. Set the initial delay for Tooltip1 to 100, for Tooltip2 set it to 1000,

When you run the program you should see "Door" the first time, and "Old and brown" pop up after.

4. Copy the label and name it lblWindow. Position LblWindow so that it is over one of the windows. Make the tooltips "Window" and "Shut tight".

Special Characters

This program uses special characters such as ü and é. To insert one of these special characters, first find its ANSI value: select **H**elp, **S**earch For Help On..., search for the word ANSI, then select ANSI Character Set. Click on the words "Characters 128-255". Scroll through the list and note the number to the left of the character you need: the letter ü is 252. To insert ü in the code hold down **ALT** and type **0252** on the **numeric** keypad. (*The codes will only work on the numeric keyboard, but will work almost anywhere in Windows, not just Visual Basic! See the table of ANSI codes.*)

Radio Buttons ☺

We could make a fairly useful program just using transparent labels and tool tips. We will modify the program so that you can use radio buttons ☺ to select English, Spanish or German for the tooltips.

1. We will not use the second tooltip for this project. You can delete it.
2. Make the form slightly larger and add 3 radio buttons. Name them RadEnglish, RadSpanish, RadGerman. Set the text to "&English", "&Spanish", "&German".
3. Make the text for the form "A House".
4. Write the code for radGerman_click as shown below:

```
Private Sub RadGerman_CheckedChanged(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles RadGerman.CheckedChanged
    Me.Text = "Ein haus"
    RadEnglish.Text = "Englisch"
    RadSpanish.Text = "Spanisch"
    RadGerman.Text = "Deutsch"
    Me.ToolTip1.SetToolTip(Me.LblDoor, "Tür") 'ü=alt+0252
    Me.ToolTip1.SetToolTip(Me.LblWindow, "Fenster")
End Sub 'RadGerman_CheckedChanged
```

Experiment: Run the program, and then try writing the code for RadEnglish before you turn the page.

```

Private Sub RadEnglish_CheckedChanged(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles RadEnglish.CheckedChanged
    Me.Text = "A house"
    RadEnglish.Text = "English"
    RadSpanish.Text = "Spanish"
    RadGerman.Text = "German"
    Me.ToolTip1.SetToolTip(Me.LblDoor, "Door")
    Me.ToolTip1.SetToolTip(Me.LblWindow, "Window")
End Sub 'RadEnglish_CheckedChanged

Private Sub RadSpanish_CheckedChanged(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles RadSpanish.CheckedChanged
    Me.Text = "Una casa"
    RadEnglish.Text = "inglés" 'é= ALT+0233
    RadSpanish.Text = "español" 'ñ= ALT+0241
    RadGerman.Text = "alemán" 'á= ALT+0225
    Me.ToolTip1.SetToolTip(Me.LblDoor, "puerta")
    Me.ToolTip1.SetToolTip(Me.LblWindow, "ventana")
End Sub 'RadSpanish_CheckedChanged

```

Save the House program and close it.

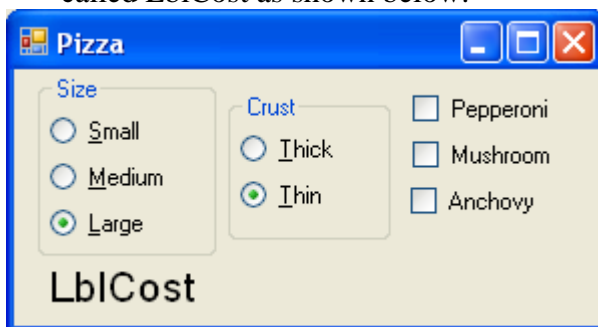
Check Boxes and Radio Buttons

Check boxes are used when the user can make several choices. Radio buttons are used when the user must select only one from a group of choices. Only one radio button in a group can be selected. In the illustration, the radio buttons are placed inside a GroupBox so that the user can select size from one group and the type of crust from another group.

1. Start a new windows application and name it Pizza.
2. Build the form as shown making sure that each radio button is drawn inside a GroupBox .
3. Name the controls GrpSize, containing RadSmall, RadMedium, and RadLarge, and GrpCrust containing RadThin and RadThick. Set the checked property to True for Large and Thin.
4. Run the program to make sure you can select one item from each group. If you can only select one button, then delete them and make sure that you draw the buttons inside the GroupBox.

We will use checkboxes for the toppings because you can have as many toppings as you want.

5. Add checkboxes named ChkPepperoni, ChkMushroom, and ChkAnchovy, and a label called LblCost as shown below:



Calculating the Cost

When any change is made, the cost of the pizza will be calculated as follows: a small pizza is \$8, medium is \$10 and large is \$12. Thick crust is \$0.50 additional and each topping is \$1

Since the cost is calculated the same way for any button that is clicked, we will create one procedure to **handle** all of the click events.

6. Double click on RadMedium to open the code window.
7. Change the name from RadSmall_Click to AnyChange.
8. Add each of the events to the handles clause: The changes are shown in bold.

```
Private Sub AnyChange(ByVal sender As Object, ByVal e As System.EventArgs) _
    Handles radMedium.Click, RadSmall.Click, RadLarge.Click, RadThick.Click, _
    RadThin.Click, ChkAnchovy.Click, ChkMushroom.Click, ChkPepperoni.Click
```

Radio buttons and CheckBoxes have a property Checked that is True when they are selected and false if not. The expression `Me.radSmall.Checked` is a Boolean expression: it does not need to be compared to True to test it.

Although we could write the statement:

```
If Me.radSmall.Checked = True Then Cost = 8
```

The statement can also be written as shown below and this is the preferred method:

```
If Me.radSmall.Checked Then Cost = 8
```

```
Private Sub AnyChange(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles RadMedium.CheckedChanged, RadSmall.Click,
    RadLarge.Click, RadThick.Click, _
    RadThin.Click, ChkAnchovy.Click, ChkMushroom.Click, ChkPepperoni.Click
    Dim Cost As Double
    If Me.RadSmall.Checked Then Cost = 8
    If Me.RadMedium.Checked Then Cost = 10
    If Me.RadLarge.Checked Then Cost = 12
    If Me.RadThick.Checked Then Cost = Cost + 0.5
    If Me.ChkAnchovy.Checked Then Cost = Cost + 1
    If Me.ChkMushroom.Checked Then Cost = Cost + 1
    If Me.ChkPepperoni.Checked Then Cost = Cost + 1
    Me.LblCost.Text = Cost
End Sub 'AnyChange
```

Run the program and save.

General Procedures

We have one procedure to handle all changes. If we would like to do something slightly different for each event we could write separate procedures for each event, but we would not want to repeat the code to calculate the cost. One important reason to not repeat code is because it can lead to errors when the code needs to be updated. We will write a general procedure to calculate cost, then call it from the other events.

Since a general procedure is not invoked by an event, type the first line under the end sub for another procedure:

```
Private Sub Calculate()
```

When we hit enter the End Sub line for Calculate is provided for us.

Move the code that calculates the price into calculate, replace it with a call to calculate:

```
Private Sub AnyChange(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles RadMedium.CheckedChanged, RadSmall.Click,
    RadLarge.Click, RadThick.Click, _
    RadThin.Click, ChkAnchovy.Click, ChkMushroom.Click, ChkPepperoni.Click
    Calculate()
End Sub 'AnyChange
Private Sub Calculate()
    Dim Cost As Double
    If Me.radSmall.Checked Then Cost = 8
    If Me.radMedium.Checked Then Cost = 10
    If Me.radLarge.Checked Then Cost = 12
    If Me.radThick.Checked Then Cost = Cost + 0.5
    If Me.chkAnchovy.Checked Then Cost = Cost + 1
    If Me.chkMushroom.Checked Then Cost = Cost + 1
    If Me.chkPepperoni.Checked Then Cost = Cost + 1
    Me.lblCost.Text = Format(Cost, "$0.00")
End Sub 'Calculate
```

Remove ChkAnchovy_Click from AnyChange and add the code below:

```
Private Sub ChkAnchovy_CheckedChanged(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles ChkAnchovy.CheckedChanged
    Me.Text = "Yuck, anchovies!"
    Calculate()
End Sub 'ChkAnchovy_CheckedChanged
```

Formatting Numbers



If you tested the program thoroughly, you should have noticed that when we select a thick crust the cost is displayed with just a 5 after the decimal place. We will solve this problem by specifying the cost be displayed with a \$ and 2 decimal places.

1. Change the last statement in the procedure to:
`Me.lblCost.Text = Format(Cost, "$0.00")`
2. Run the program and try several selections including thick crust to make sure that the program works.
3. Double click on the word "Format" to select it, then press F1 for help. Look at the examples and explanation. Several examples refer to formatting the date.

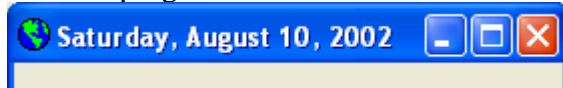
Experiment: Add another size, Extra Large, and add a few more toppings.

Formatting Dates

1. Save the Pizza program, then close it and start a new Windows application called Today.
2. Double click the form and add the line shown below to the form load event:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
    Me.Text = Format(Now(), "D")
End Sub
```

3. Run the program to see the date as shown below:



Experiment: Try each of the formats shown below:

```
Me.Text = Format(Now(), "m") 'August 10
Me.Text = Format(Now(), "Long Time") '11:23:16 AM
Me.Text = Format(Now(), "ddd") 'Sat
Me.Text = Format(Now(), "hh:mm") '11:34
```

You can also format a date other than today. What day were you born? Declare your birthday as shown below:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
    Dim MyBirthday As Date = #5/21/1976#
    Me.Text = Format(MyBirthday, "long date")
End Sub
```

