**Tutorial on Toolbars, List, ListView**

**Create an Application**

1. Run your Microsoft Visual Express for VB or C# 2010
2. Click New Project (or Create Project)



1. Name your Project Application to **ToolbarDemo**.
2. Select your Form and Open the Property Window and name your *Form* to *frmMain* and change the value of *Text* property to *Toolbar Demo*.





1. Change the *Maximize Box* property value to False. And *Width* under *Size* property to **570** and *Height* to **440**.





1. Save your work first by clicking on the Save All icon in the Toolbar as shown below or press *Ctrl+Shift+S*.



1. Go to your *Toolbox* and under *Menus and Toolbar* drag and drop *Toolstrip* to your form.



1. Change the name of Tool Strip in your Property Window from *ToolStrip1* to *tsMain*.



1. Select the *ToolStrip* in your Form and then click the Drop down menu and Choose *Button* as shown below.



1. Go to [www.iconarchive.com](http://www.iconarchive.com) and in their *Category* menu select *System Icons*. And then click *Windows Business* as shown below.



1. Select *My Documents* icon and then later on *Network icon* as well. Download the 96px x 96px icons as shown below.



1. Let’s go back to Visual Studio and select the Tool strip button that we created awhile ago. Change the *(Name)* property first to tsbFolder and *Text* property to *Open Folder*.





1. Click the button in *Image* property



1. Under *Select Resource* click Import and then select the *My Documents* icon that you downloaded awhile ago and then click *Ok* button.



1. Go to *Image Scaling* property and select *None*.



1. Go to your Toolbar drop down and select *Separator* as shown below.



1. Go to Toolbar drop down again and this time select *Button*.
2. Repeat Steps 11 to 14 and but the name of your Tool bar strip should be *tsbInternet* instead of *tsbFolder* and the *Text* property should be *Go to Internet*. And instead of choosing My Documents icon choose *Network* icon and it should look like one below.



1. Run (F5) your application and point and then click on any icon. As you could see you could not yet show that the icon is selected when you click it so far. That would change next.
2. Close the Form that you run not the Visual Studio. And change both the *CheckOnClick* property to True of *tsbFolder* and *tsbInternet*.
3. Run again and you could see that you could select both the tsbFolder and tsbInternet Tool strip buttons. However, as you can see you could select both but it does not toggle.



1. So, what we are going to do this time is to let this one toggle. So, close again the Form that you run. Go to tsbFolder first and change the *CheckedState* to *Checked*.



1. Double click ton tsbFolder and you will see the code below:

**VB.Net**:



**C#:**



1. What we are going to do is add the code to let it Toggle.

**VB.Net**:

Private Sub tsbFolder\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbFolder.Click

 If (tsbFolder.Checked = True) Then

 tsbInternet.Checked = False

 End If

End Sub

**C#:**

 private void tsbFolder\_Click(object sender, EventArgs e)

 {

 if (tsbFolder.Checked)

 {

 tsbInternet.Checked = false;

 }

 }

1. Go back to Form1 Design



1. Double click on tsbInternet this time follow the below.

**VB.Net**:

Private Sub tsbInternet\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbInternet.Click

 If (tsbInternet.Checked = True) Then

 tsbFolder.Checked = False

 End If

 End Sub

**C#:**

 private void tsbInternet\_Click(object sender, EventArgs e)

 {

 if (tsbInternet.Checked)

 {

 tsbFolder.Checked = false;

 }

 }

1. Run your application again and see what happens.

**Adding a TabControl**

A TabControl allows you to create multiple interfaces in your Form organized according to your own liking.

1. Go to *ToolBox* under *Containers* drag and drop *TabControl* in your Form.



1. Change the *Size* property of your *TabControl* with the following values: *Width* = 538, *Height* = 288.
2. Change the *(Name)* of your *TabControl* to *tcMain*.



1. Click on the body of the *TabPage*

**

1. And change the name *TabPage1* to *tpFolder* and the *Text* property to *Folder*





1. Do the same to TabPage2 using the following values :

**(Name)** : tpNetwork

**Text**  : Network



1. Add a *CheckedListBox* under *Common Controls* in your *Toolbox* and drop it on the *Form*.



1. Change the *(Name)* of the *CheckedListBox* that you have just dropped from *CheckedListbox1* to *chkFolder*.



1. Go to *Items* property and click it and add the following: Videos, School, Pictures and Docs





1. Get a *Button* from the Toolbox and drop it to the form near to the right of the *CheckedListBox* that you have created awhile ago.



1. Change the name of the Button from *Button1* to *btnAddItem*. Change also the Button1 value in Text property to >> and then change the Size 🡪Width property of the Button to 27.



1. Add a *ListView* from the Toolbox and drop it to the *Form*.



1. Change the (Name) from ListView1 to *lvwItemsBought* and change the Size🡪Width=328,Height=229.



1. Click on the *ListView* that you have just created particularly on the arrow on the top right that will show you a dropdown box like below:



1. Change View from *LargeIcon* to *Details*



1. Click Edit *Columns* and then click Add button.
2. Change (Name) property from *ColumnHeader1* to *colFolderName* and *Width* from 60 to 200 and then *Text* property from *ColumnHeader* to *FolderName*.





1. Add Another column with the following values:

**(Name)** : colCount

**Text**  : File Count

**Width** : 110

****

1. Now, we are going to add whatever is checked from the *CheckedListBox* to the *ListView*. First, double click on the btnAddItem and you should have the code behind like one below:

**VB.Net**:

 Private Sub btnAddItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAddItem.Click

 End Sub

**C#:**

 private void btnAddItem\_Click(object sender, EventArgs e)

 {

 }

1. What we are going to do here is first check if the user checked any checkbox from our *CheckedListBox*. To do that we need to loop first using a for next loop in order to loop through the CheckedListBox that is checked by the User. But before that *CheckedItems* property is an array and .Net array index always starts with zero (0).

**VB.Net**:

 Private Sub btnAddItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAddItem.Click

For i As Integer = 0 To chkFolder.CheckedItems.Count

 Next

 End Sub

**C#:**

private void btnAddItem\_Click(object sender, EventArgs e)

{

 for (int i = 0; i < chkFolder.CheckedItems.Count; i++)

 {

 }

}

1. Then we are going to add the checked Item to the *ListView* that we also created awhile ago. But before that I’ll give you some information on *ListView*. The first column on your ListView is the *Item* and the succeeding columns (2nd and more) are what are known as the *SubItems*. But first let’s add the Item and will add the *SubItem* later on.

**VB.Net**:

Private Sub btnAddItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAddItem.Click

 For i As Integer = 0 To chkFolder.CheckedItems.Count - 1

 Dim itm As ListViewItem

 Dim chkItem As String

 Dim cnt As Integer = 0

 chkItem = chkFolder.CheckedItems(i).ToString()

 itm = lvwItemsBought.Items.Add(chkItem)

 Next

End Sub

**C#:**

private void btnAddItem\_Click(object sender, EventArgs e)

{

 for (int i = 0; i < chkFolder.CheckedItems.Count; i++)

 {

 ListViewItem itm;

 String chkItem;

 int cnt = 0;

 chkItem = chkFolder.CheckedItems[i].ToString();

 itm = lvwItemsBought.Items.Add(chkItem);

 }

}

1. Run your program and see checked some Items and then click the button >> to add the items that you checked into the *ListView*.
2. Now, we are going to add the *Count* column which is a *SubItem* in *ListView*. But before that we are going to use *if..else* condition in order add the SubItem. For example if the user checked Videos we will add 5, if School then 10, if Pictures 20 and if Docs 7.

**VB.Net**:

Private Sub btnAddItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAddItem.Click

 For i As Integer = 0 To chkFolder.CheckedItems.Count - 1

 Dim itm As ListViewItem

 Dim chkItem As String

 Dim cnt As Integer = 0

 chkItem = chkFolder.CheckedItems(i).ToString()

 itm = lvwItemsBought.Items.Add(chkItem)

 If (chkItem = "Videos") Then

 cnt = 5

 ElseIf (chkItem = "School") Then

 cnt = 10

 ElseIf (chkItem = "Pictures") Then

 cnt = 20

 ElseIf (chkItem = "Docs") Then

 cnt = 7

 End If

 Next

End Sub

**C#:**

private void btnAddItem\_Click(object sender, EventArgs e)

{

 for (int i = 0; i < chkFolder.CheckedItems.Count; i++)

 {

 ListViewItem itm;

 String chkItem;

 int cnt = 0;

 chkItem = chkFolder.CheckedItems[i].ToString();

 itm = lvwItemsBought.Items.Add(chkItem);

 if (chkItem == "Videos")

 {

 cnt = 5;

 }

 else if (chkItem == "School")

 {

 cnt = 10;

 }

 else if (chkItem == "Pictures")

 {

 cnt = 20;

 }

 else if (chkItem == "Docs")

 {

 cnt = 7;

 }

 }

}

1. Now we are going to add the cnt or the SubItem finally this time.

**VB.Net**:

Private Sub btnAddItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAddItem.Click

 For i As Integer = 0 To chkFolder.CheckedItems.Count - 1

 Dim itm As ListViewItem

 Dim chkItem As String

 Dim cnt As Integer = 0

 chkItem = chkFolder.CheckedItems(i).ToString()

 itm = lvwItemsBought.Items.Add(chkItem)

 If (chkItem = "Videos") Then

 cnt = 5

 ElseIf (chkItem = "School") Then

 cnt = 10

 ElseIf (chkItem = "Pictures") Then

 cnt = 20

 ElseIf (chkItem = "Docs") Then

 cnt = 7

 End If

 itm.SubItems.Add(cnt)

 Next

End Sub

**C#:**

private void btnAddItem\_Click(object sender, EventArgs e)

{

 for (int i = 0; i < chkFolder.CheckedItems.Count; i++)

 {

 ListViewItem itm;

 String chkItem;

 int cnt = 0;

 chkItem = chkFolder.CheckedItems[i].ToString();

 itm = lvwItemsBought.Items.Add(chkItem);

 if (chkItem == "Videos")

 {

 cnt = 5;

 }

 else if (chkItem == "School")

 {

 cnt = 10;

 }

 else if (chkItem == "Pictures")

 {

 cnt = 20;

 }

 else if (chkItem == "Docs")

 {

 cnt = 7;

 }

 itm.SubItems.Add(cnt.ToString());

 }

}

1. Run your program an again checked as many items as you want in your *CheckBoxList* and then press the >> button.