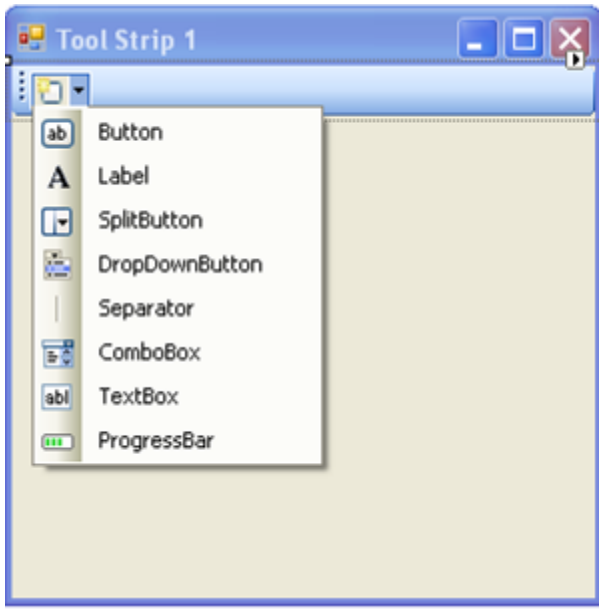
Tool Strips, Status Strips, and Splitters

Tool Strips

- Usually called *tool bars*.
- Create easily customized, commonly employed toolbars that support advanced user interface and layout features, such as docking, rafting, buttons with text and images, drop-down buttons and controls, overflow buttons, and run-time reordering of *ToolStrip* items.
- Support the typical appearance and behavior of the operating system.
- Handle events consistently for all containers and contained items, in the same way you handle events for other controls.
- Drag items from one *ToolStrip* to another or within a *ToolStrip*.
- Create drop-down controls and user interface type editors with advanced layouts in a *ToolStripDropDown*.

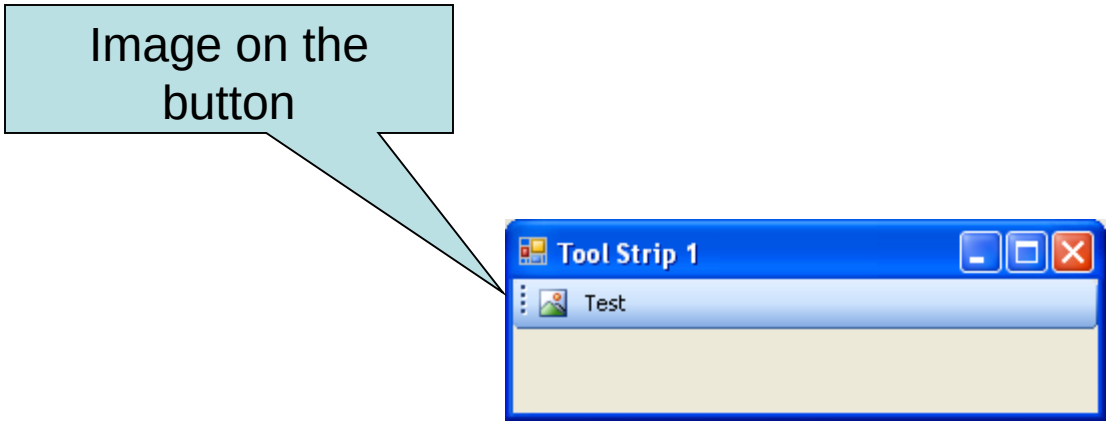
Tool Strip Item Selection



Buttons

- You have three options regarding the image:
  1. Display text instead of the image.
  2. Display a different image.
  3. Display both text and an image.
  4. Don't display anything. This results in an empty box.

Using an Image



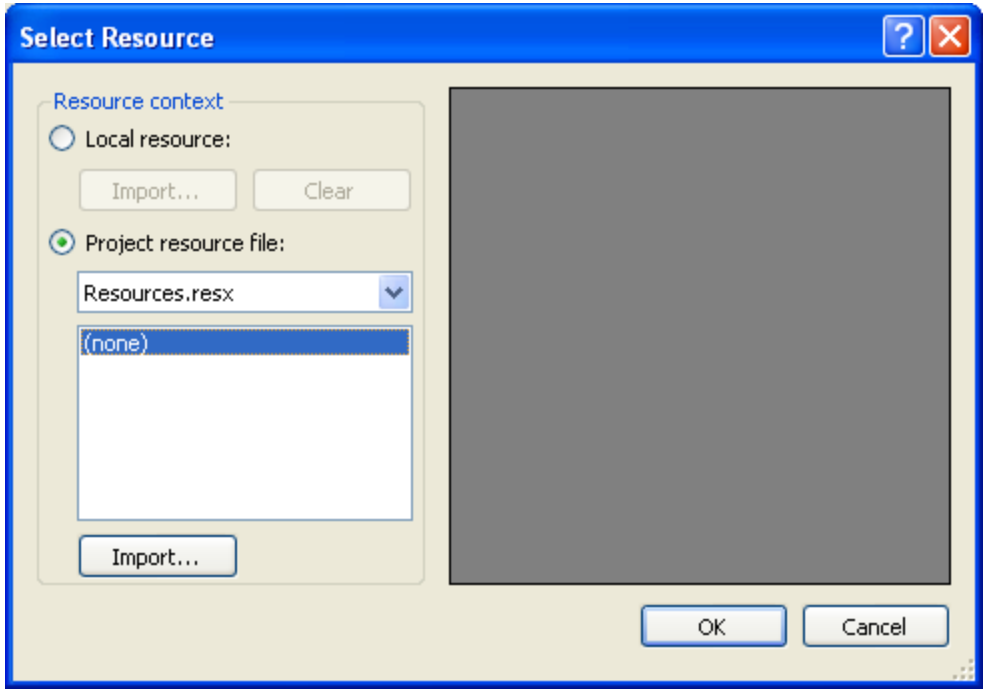
The Visual Studio Image Library

- Get it from here.
- Unpack this file using whatever tool you prefer or right click the file name and select extract.
- The images used for tool strip buttons are normally 16 x 16 pixels.
- Use the *ImageScaling* property to set the size if not 16 x 16.

Adding the Image

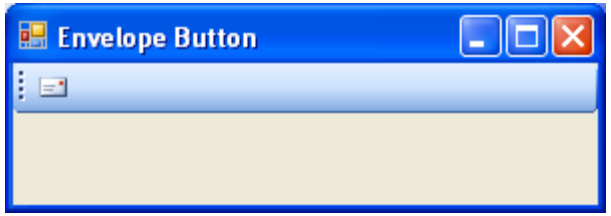
- To get going just click the ellipses for the *Image* property of the *ToolStripButton* you have selected in the designer.
- You have two choices as to where your image will be stored:
  1. Local resource - placed in the file <formname>.resx as a hexadecimal encoding.
  2. Project resource file - placed in the file Properties\Resources.resx as a reference to the file which is copied to the Resources folder.

Select Resource Dialog



Example

- Here is an example of a *ToolStripButton* with the image *Envelope.bmp* which I imported from the Visual Studio Image Library:

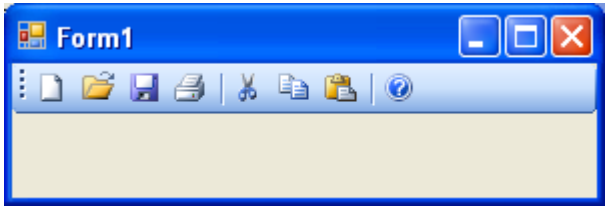


Adding Standard Buttons

- A very nice feature of the designer allows you to add a group of standard buttons with only a couple of mouse clicks.
- After you have created the *ToolStrip* control click on the small arrow in the upper right hand corner of the control.
- The *ToolStrip* task pane will appear. Choose *Insert Standard Items*.

Resulting Order

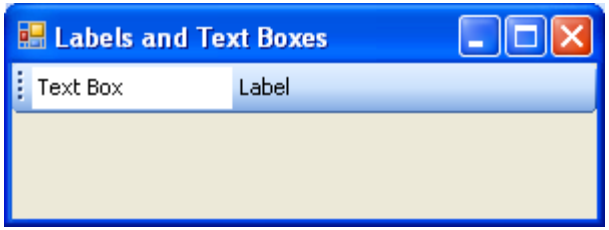
1. New
2. Open
3. Save
4. Print
5. Cut
6. Copy
7. Paste
8. Help



Labels and Text Boxes

- These are represented by the *ToolStripLabel* and *ToolStripTextBox* classes.
- The appearance is different and only text boxes can be edited by the user when the application runs.
- The *Text* property of each control is used to set the text to be displayed.

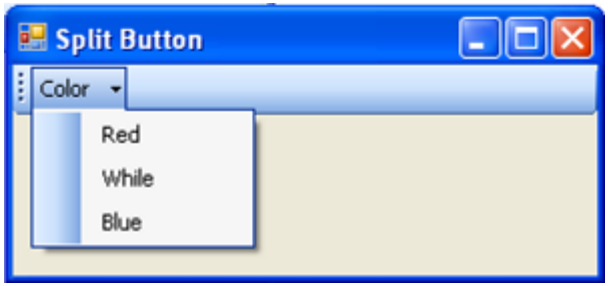
Example



Drop Down Buttons and Split Buttons

- The major difference between these two buttons and the *ToolStripMenuItem* is that they are both displayed to look like a button.
- Each one has an image, text, or both as well as a drop down list of text items.

Example



Combo Box Buttons

- The *ToolStripComboBox* control is one of the most popular found in tool bars in current applications.
- It consists of a text box combined with a drop down list to make a selection.
- The *DropDownStyle* property can be set to one of the selections shown in the following table.

Drop Down Styles

Style	*Description*
DropDown	The text portion is editable. The user must click the arrow button to display the list portion. This is the default style.
DropDownList	The user cannot directly edit the text portion. The user must click the arrow button to display the list portion.
Simple	The text portion is editable. The list portion is always visible.

Combo Box Example

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
namespace ComboButton1
{
    public partial class Form1 : Form
    {
        private Font myFont;
```

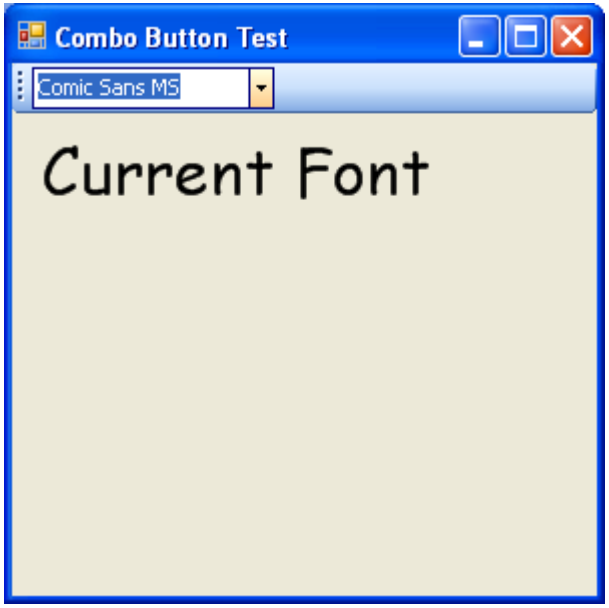
Combo Box Example

```
public Form1()
{
    InitializeComponent();
    toolStripComboBox1.SelectedIndex = 0;
}
protected override void OnPaint(PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.DrawString("Current Font", myFont,
          Brushes.Black, 10, 30);
```

Combo Box Example

```
        }
        private void toolStripComboBox1_SelectedIndexChanged(object sender,
             EventArgs e)
        {
            myFont = new Font(toolStripComboBox1.Text,
                  24);
            Invalidate();
        }
    }
}
```

Combo Box Example



Progress Bars

- Very similar to the progress bar control.
- After you add it to your *ToolStrip* you should set the *Maximumum* property to the number of increments you want.
- You should also change the *Step* property to the number of increments you want each time you call the *PerformStep* method.
- All you need to do is place a call to *PerformStep* in the appropriate point in the time consuming operation.
- The following example uses a timer.
- Remember not to tie up the UI thread! Use a worker thread when appropriate.

Progress Bar Example

```
ProgressBar1 - Form1.cs
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
namespace ProgressBar1
{
    public partial class Form1 : Form
    {
```

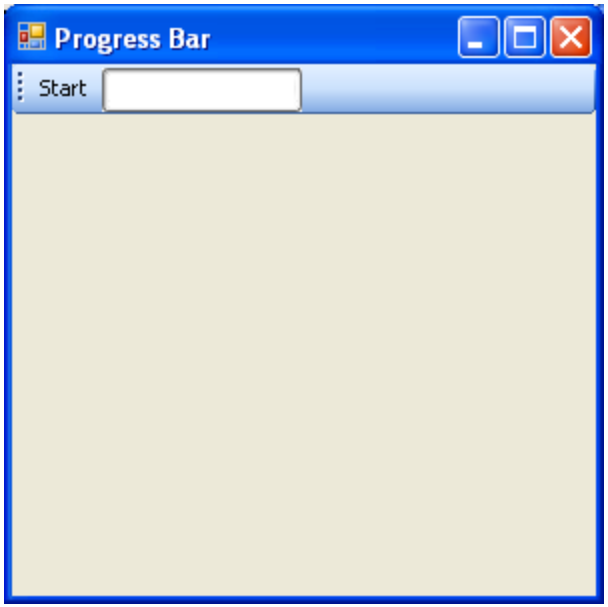
Progress Bar Example

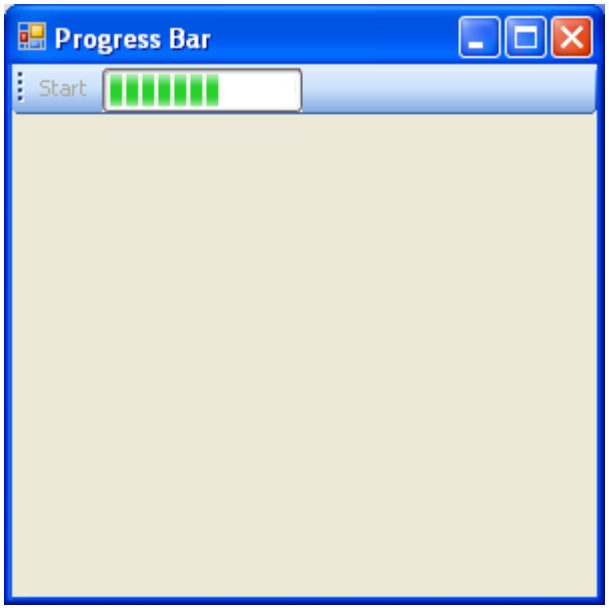
```
public Form1()
{
    InitializeComponent();
}
private void toolStripButton1_Click(object sender,
     EventArgs e)
{
    toolStripButton1.Enabled = false;
    timer1.Enabled = true;
}
private void timer1_Tick(object sender,
    EventArgs e)
{
```

Progress Bar Example

```
            pbar.PerformStep();
            if (pbar.Value >= pbar.Maximum)
            {
                timer1.Enabled = false;
                pbar.Value = pbar.Minimum;
                toolStripButton1.Enabled = true;
            }
        }
    }
}
```

Progress Bar Example





Status Strips

- We can add the following items:
  1. StatusLabel
  2. ProgressBar
  3. DropDownButton
  4. SplitButton

ToolStripStatusLabel Properties (partial list)

Property	Description
Autosize	*Determines if the item should automatically size to its content.*
Size	*The size if autosize is not used.*
Text	*Set this property to display the text you want.*
TextAlign	*Controls text alignment, e.g., left, right, center, etc.*
Visible	*Use this property to programmaticallyshow or hide the label.*

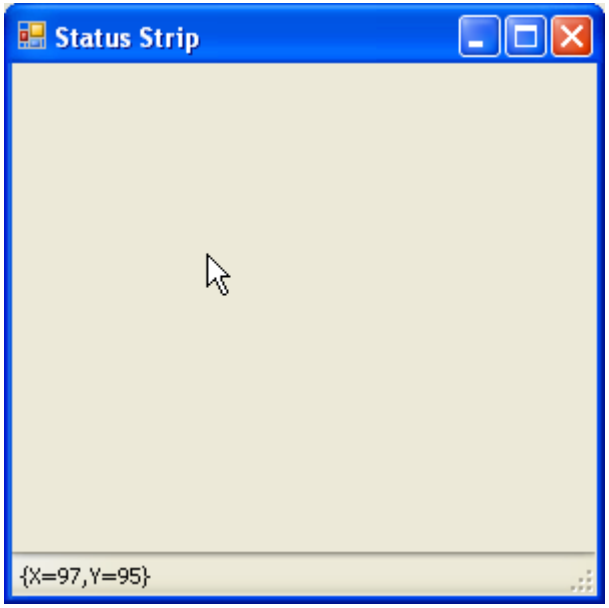
Status Bar Example

```
StatusStrip1 - Form1.cs
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
namespace StatusStrip1
{
    public partial class Form1 : Form
    {
```

Status Bar Example

```
        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_MouseMove(object sender,
MouseEventArgs e)
        {
            position.Text = e.Location.ToString();
        }
    }
}
```

Status Bar Example



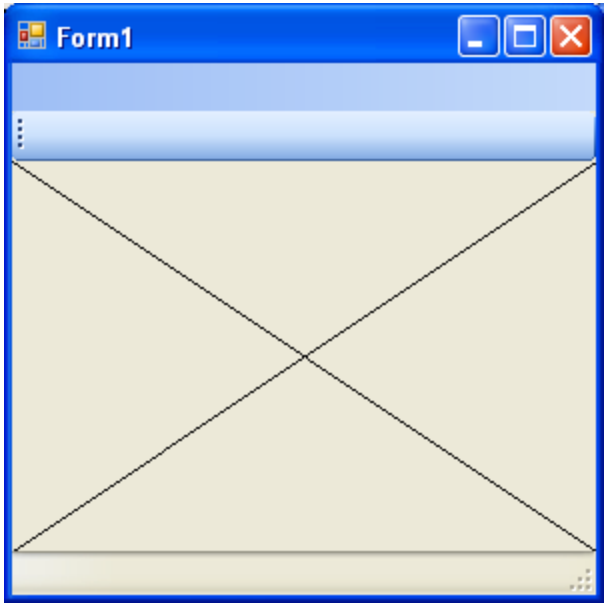
Docking Controls

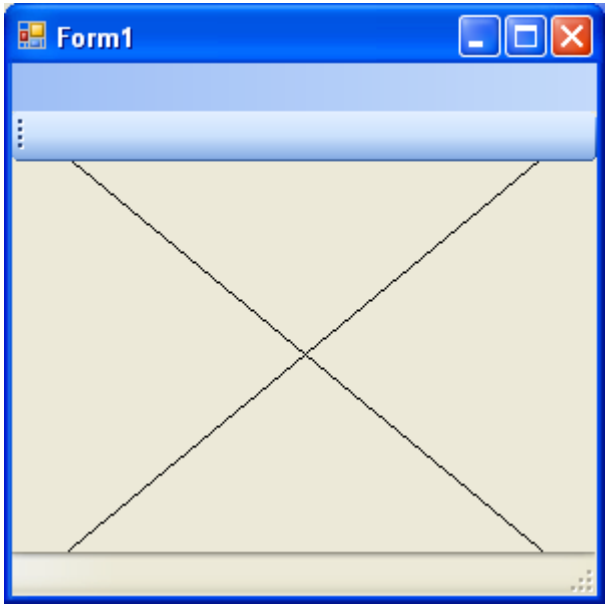
- The order we add the controls determines the overlap.
- The following example shows two possible orders.
- Right click the panel and select *Bring to Front* to force the correct order.
- The Z order of controls can be established this way.

Docking Controls

```
private void panel1_Paint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.DrawLine(Pens.Black, 0, 0, panel1.Width,
        panel1.Height);
    g.DrawLine(Pens.Black, panel1.Width, 0, 0,
        panel1.Height);
}
```

Two Z Orders

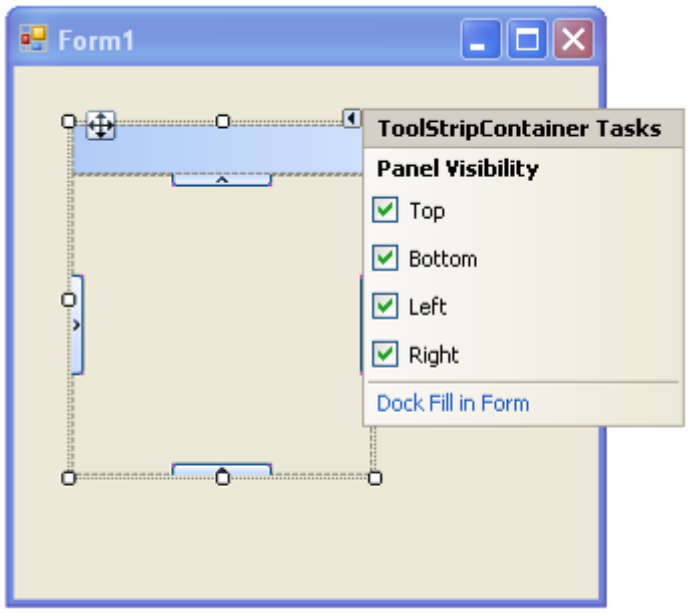




Tool Strip Containers

- If you drag a *ToolStripContainer* to your form an options list will appear as shown in the next slide.
- Uncheck the panels you don't want and then click on *Dock Fill in Form*.

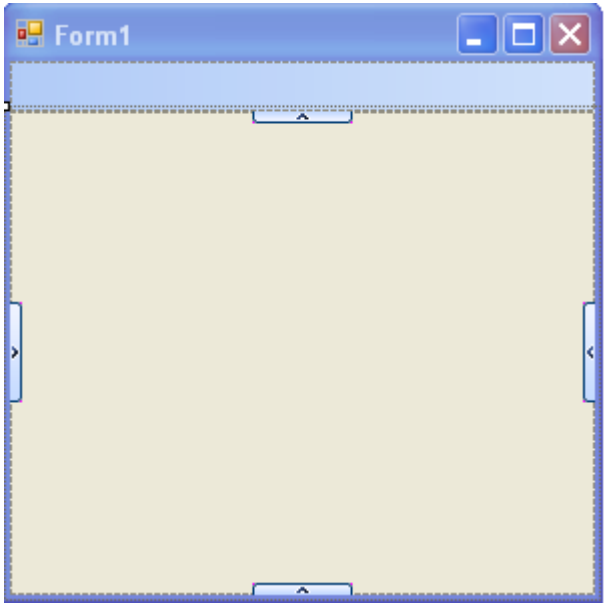
Setting the Options



Adding Controls

- Open any of the edges and drag the controls you want.
- *Rafting* is automatically enabled.

Opening the Top Container



The Panel Properties

Property	*Description*
ContentPanel	The central panel for your form.
TopToolStripPanel	The top panel.
BottomToolStripPanel	The bottom panel.
LeftToolStripPanel	The left panel.
RightToolStripPanel	The right panel.

Rafting