Floating Toolbar


```
/*
User Interfaces in C#: Windows Forms and Custom Controls
by Matthew MacDonald

Publisher: Apress
ISBN: 1590590457
*/

using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace FloatingToolbar
{
///

/// Summary description for FloatingToolbar.
///
public class FloatingToolbar : System.Windows.Forms.Form
{
internal System.Windows.Forms.ImageList imgButtons;
internal System.Windows.Forms.ToolBar toolBar1;
internal System.Windows.Forms.ToolBarButton cmdNew;
internal System.Windows.Forms.ToolBarButton cmdOpen;
internal System.Windows.Forms.ToolBarButton cmdClose;
internal System.Windows.Forms.ToolBarButton cmdSave;
internal System.Windows.Forms.ToolBarButton ToolBarButton1;
internal System.Windows.Forms.ToolBarButton cmdPreview;
private System.ComponentModel.IContainer components;
```

```
public FloatingToolbar()
{
//
// Required for Windows Form Designer support
//
InitializeComponent();

//
// TODO: Add any constructor code after InitializeComponent
call
//
}

///

/// Clean up any resources being used.
///
protected override void Dispose( bool disposing )
{
if( disposing )
{
if (components != null)
{
components.Dispose();
}
}
base.Dispose( disposing );
}

#region Windows Form Designer generated code
///

/// Required method for Designer support – do not modify
/// the contents of this method with the code editor.
///
private void InitializeComponent()
{
this.components = new System.ComponentModel.Container();
System.Resources.ResourceManager resources = new
```

```
System.Resources.ResourceManager(typeof(FloatingToolbar));
this.imgButtons = new
System.Windows.Forms.ImageList(this.components);
this.toolBar1 = new System.Windows.Forms.ToolBar();
this.cmdNew = new System.Windows.Forms.ToolBarButton();
this.cmdOpen = new System.Windows.Forms.ToolBarButton();
this.cmdClose = new System.Windows.Forms.ToolBarButton();
this.cmdSave = new System.Windows.Forms.ToolBarButton();
this.ToolBarButton1 = new
System.Windows.Forms.ToolBarButton();
this.cmdPreview = new System.Windows.Forms.ToolBarButton();
this.SuspendLayout();
//
// imgButtons
//
this.imgButtons.ColorDepth =
System.Windows.Forms.ColorDepth.Depth8Bit;
this.imgButtons.ImageSize = new System.Drawing.Size(16, 16);
this.imgButtons.ImageStream =
((System.Windows.Forms.ImageListStreamer)(resources.GetObject(
“imgButtons.ImageStream”)));
this.imgButtons.TransparentColor =
System.Drawing.Color.Transparent;
//
// toolBar1
//
this.toolBar1.Appearance =
System.Windows.Forms.ToolBarAppearance.Flat;
this.toolBar1.BorderStyle =
System.Windows.Forms.BorderStyle.Fixed3D;
this.toolBar1.Buttons.AddRange(new
System.Windows.Forms.ToolBarButton[] {
this.cmdNew,
this.cmdOpen,
this.cmdClose,
this.cmdSave,
this.ToolBarButton1,
```

```
this.cmdPreview});
this.toolBar1.DropDownArrows = true;
this.toolBar1.ImageList = this.imgButtons;
this.toolBar1.Name = “toolBar1”;
this.toolBar1.ShowToolTips = true;
this.toolBar1.Size = new System.Drawing.Size(292, 41);
this.toolBar1.TabIndex = 5;
this.toolBar1.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.toolBar1_MouseUp);
this.toolBar1.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.toolBar1_MouseMove
);
this.toolBar1.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.toolBar1_MouseDown
);
//
// cmdNew
//
this.cmdNew.ImageIndex = 3;
this.cmdNew.Text = “New”;
//
// cmdOpen
//
this.cmdOpen.ImageIndex = 0;
this.cmdOpen.Text = “Open”;
//
// cmdClose
//
this.cmdClose.ImageIndex = 1;
this.cmdClose.Text = “Close”;
//
// cmdSave
//
this.cmdSave.ImageIndex = 2;
this.cmdSave.Text = “Save”;
//
// ToolBarButton1
```

```
//
this.ToolBarButton1.Style                                    =
System.Windows.Forms.ToolBarButtonStyle.Separator;
//
// cmdPreview
//
this.cmdPreview.ImageIndex = 4;
this.cmdPreview.Text = “Preview”;
//
// FloatingToolbar
//
this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.ClientSize = new System.Drawing.Size(292, 266);
this.Controls.AddRange(new System.Windows.Forms.Control[] {
this.toolBar1});
this.IsMdiContainer = true;
this.Name = “FloatingToolbar”;
this.Text = “Floating Toolbar”;
this.ResumeLayout(false);

}
#endregion

///

/// The main entry point for the application.
///
[STAThread]
static void Main()
{
Application.Run(new FloatingToolbar());
}

private bool draggingToolbar;
private Point draggedFrom;

private   void   toolBar1_MouseMove(object   sender,
System.Windows.Forms.MouseEventArgs e)
```

```
{
if (draggingToolbar)
{
//if (toolBar1.Dock == DockStyle.Top || toolBar1.Dock ==
DockStyle.Left)
if (toolBar1.Dock == DockStyle.Top)
{
// Check it the dragging has reached the threshold.
if (draggedFrom.X < (e.X - 20) || draggedFrom.Y < (e.Y - 20))
{ draggingToolbar = false; // Disconnect the toolbar.
toolBar1.Dock = DockStyle.None; toolBar1.Location = new
Point(10, 10); toolBar1.Size = new Size(200, 100);
toolBar1.BorderStyle = BorderStyle.FixedSingle; } } else if
(toolBar1.Dock == DockStyle.None) { toolBar1.Left = e.X +
toolBar1.Left - draggedFrom.X; toolBar1.Top = e.Y +
toolBar1.Top - draggedFrom.Y; if (toolBar1.Top < 5) {
draggingToolbar = false; // Re-dock the control. toolBar1.Dock
= DockStyle.Top; toolBar1.BorderStyle = BorderStyle.None; }
else if (toolBar1.Left < 5) { draggingToolbar = false; // Re-
dock the control. toolBar1.Dock = DockStyle.Left;
toolBar1.BorderStyle = BorderStyle.None; } } } } private void
toolBar1_MouseDown(object sender,
System.Windows.Forms.MouseEventArgs e) { draggingToolbar =
true; draggedFrom = new Point(e.X, e.Y); toolBar1.Capture =
true; } private void toolBar1_MouseUp(object sender,
System.Windows.Forms.MouseEventArgs e) { draggingToolbar =
false; toolBar1.Capture = false; } } } FloatingToolbar.zip( 30
k)[/csharp]
```