

www.bytescout.com

How to create full featured screen capturing app in C# with ByteScout Screen Capturing SDK

The tutorial shows how to create full featured screen capturing app in C#

These sample source codes on this page below are demonstrating how to create full featured screen capturing app in C#. Want to create full featured screen capturing app in your C# app? ByteScout Screen Capturing SDK is designed for it. ByteScout Screen Capturing SDK is the tool for developers who want to add screen capturing in their application. Can record screen into video and into single screenshots. Output formats are WMV, AVI, WebM for video and PNG for screenshots. You can adjust output video size, quality, resolution, framerate, video and audio codecs. Includes special privacy features for blacking out sensitive information on screen. Can also capture video from web camera, can add overlays with text or images.

This code snippet below for ByteScout Screen Capturing SDK works best when you need to quickly create full featured screen capturing app in your C# application. In order to implement the functionality, you should copy and paste this code for C# below into your code editor with your app, compile and run your application. Detailed tutorials and documentation are available along with installed ByteScout Screen Capturing SDK if you'd like to dive deeper into the topic and the details of the API.

Free trial version of ByteScout Screen Capturing SDK is available on our website. Documentation and source code samples are included.

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about ByteScout Screen Capturing SDK](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Screen Capturing SDK](#)

[Get Free API key for Web API](#)

[visit www.ByteScout.com](#)

Source Code Files:

CapturingThread.cs

```
using System;
using System.Drawing;
using System.IO;
using System.Runtime.InteropServices;
using System.Threading;
using System.Windows.Forms;
using BytescoutScreenCapturingLib;

// NOTE: if you are getting error like "invalid image" related to loading the SDK's DLL
// try to do the following:
// 1) remove the reference to the SDK by View - Solution Explorer
// then click on References, select Bytescout... reference name and right-click it and
// 2) To re-add click on the menu: Project - Add Reference
// 3) In "Add Reference" dialog switch to "COM" tab and find Bytescout...
// 4) Select it and click "Add"
// 5) Recompile the application
// Note: if you need to run on both x64 and x86 then please make sure you have set "Embed
// in Assembly" option for the BytescoutScreenCapturingLib.dll reference

namespace ScreenCapturing
{
    public class CapturingThreadData
    {
        public CaptureAreaType CaptureType;
        public String TempFile;
        public Rectangle CaptureRectangle;
        public bool ShowWebCamStream;

        public int Result = 0; // 0 - success; 1 - error
        public string ErrorText;
    }

    public class CapturingThread
    {
        public static void ThreadProc(Object obj)
        {
            Capturer capturer = new Capturer(); // create new screen capturer object

            CapturingThreadData data = (CapturingThreadData) obj;

            if (Program.Cfg.WriteLog)
                capturer.SetLogFile(Path.GetTempPath() + Application.ProductName + ".log");

            capturer.RegistrationName = "demo";
            capturer.RegistrationKey = "demo";

            if (Program.Cfg.AudioDevice != "")
            {
                capturer.CurrentAudioDeviceName = Program.Cfg.AudioDevice;
            }

            if (Program.Cfg.AudioLine != "")
            {
                capturer.CurrentAudioLineName = Program.Cfg.AudioLine;
            }
        }
    }
}
```

```

        capturer.CurrentAudioDeviceLineName = Program.Cfg.AudioLine;
    }

    if (Program.Cfg.SelectedVideoCodecTab == 0)
    {
        capturer.CurrentWMVAudioCodecName = Program.Cfg.WmvAudioCodecName;
        capturer.CurrentWMVAudioFormat = Program.Cfg.WmvAudioFormat;
        capturer.CurrentWMVVideoCodecName = Program.Cfg.WmvVideoCodecName;

        Program.Cfg.WmvAudioCodec = capturer.CurrentWMVAudioCodec;
        Program.Cfg.WmvAudioFormat = capturer.CurrentWMVAudioFormat;
        Program.Cfg.WmvVideoCodec = capturer.CurrentWMVVideoCodec;
    }
    else
    {
        capturer.CurrentAudioCodecName = Program.Cfg.AviAudioCodecName;
        capturer.CurrentVideoCodecName = Program.Cfg.AviVideoCodecName;
    }

    capturer.AudioEnabled = Program.Cfg.EnableAudio;
    // this option tells to use captured area dimensions as output video width,
    // or use user defined video dimensions
    capturer.MatchOutputSizeToTheSourceSize = !Program.Cfg.ResizeOutputVideo;
    capturer.FPS = Program.Cfg.FPS;

    capturer.ShowMouseHotSpot = Program.Cfg.ShowMouseHotSpot;
    capturer.CaptureMouseCursor = Program.Cfg.CaptureMouseCursor;
    capturer.AnimateMouseClicks = Program.Cfg.AnimateMouseClicks;
    capturer.AnimateMouseButtons = Program.Cfg.AnimateMouseButtons;
    capturer.MouseAnimationDuration = Program.Cfg.MouseAnimationDuration;
    capturer.MouseSpotRadius = Program.Cfg.MouseSpotRadius;
    capturer.MouseHotSpotColor = (uint) ColorTranslator.ToOle(Program.Cfg.MouseHotSpotColor);
    capturer.MouseCursorLeftClickAnimationColor = (uint) ColorTranslator.ToOle(Program.Cfg.MouseCursorLeftClickAnimationColor);
    capturer.MouseCursorRightClickAnimationColor = (uint) ColorTranslator.ToOle(Program.Cfg.MouseCursorRightClickAnimationColor);

    capturer.CaptureRectLeft = data.CaptureRectangle.Left;
    capturer.CaptureRectTop = data.CaptureRectangle.Top;
    capturer.CaptureRectWidth = data.CaptureRectangle.Width;
    capturer.CaptureRectHeight = data.CaptureRectangle.Height;

    capturer.KeepAspectRatio = Program.Cfg.KeepAspectRatio;

    // show recording time stamp
    capturer.OverlayingRedTextCaption = "Recording: {RUNNINGMIN}:{RUNNINGSEC} :{FRAMERATE} FPS:{FRAMERATE} Fps";

    capturer.OutputWidth = Program.Cfg.OutputWidth;
    capturer.OutputHeight = Program.Cfg.OutputHeight;

    if ((capturer.WebCamCount > 0) && (data.ShowWebCamStream))
    {
        capturer.AddWebCamVideo = true;

        if (!String.IsNullOrEmpty(Program.Cfg.WebCameraDevice))
        {
            capturer.CurrentWebCamName = Program.Cfg.WebCameraDevice;
        }

        capturer.SetWebCamVideoRectangle(Program.Cfg.WebCameraDevice);
    }
}

```

```
        data.TempFile = Path.GetTempFileName();
        data.TempFile = Path.ChangeExtension(data.TempFile, (Program.C
capturer.OutputFileName = data.TempFile;
capturer.CapturingType = data.CaptureType;

// set border around captured area if we are not capturing entire screen
if (capturer.CapturingType != CaptureAreaType.catScreen &&
    capturer.CapturingType != CaptureAreaType.catWebcamFullScreen)
{
    // set border style
    capturer.CaptureAreaBorderType = Program.Cfg.CaptureAre
    capturer.CaptureAreaBorderColor = (uint) ColorTranslat
    capturer.CaptureAreaBorderWidth = Program.Cfg.CaptureA
}

try
{
    capturer.Run();

// IMPORTANT: if you want to check for some code if need to stop the recording
// using Thread.Sleep(1) inside the checking loop, so you have the loop like
// Do
// Thread.Sleep(1)
// While StopButtonNotClicked

}
catch (COMException ex)
{
    data.ErrorText = ex.Message;
    data.Result = 1;
    Marshal.ReleaseComObject(capturer);
    return;
}

try
{
    Thread.Sleep(Timeout.Infinite);
}
catch (ThreadInterruptedException)
{
    capturer.Stop();
    data.Result = 0;
}
catch (Exception ex)
{
    data.ErrorText = ex.Message;
    data.Result = 1;
}
finally
{
    Marshal.ReleaseComObject(capturer);
}
}

}

}
```

ColorControl.cs

```
using System;
using System.Drawing;
using System.Windows.Forms;
using System.ComponentModel;

namespace ScreenCapturing
{
    public enum CustomBorderStyle { None, Dashed, Dotted, Bump, Etched, Flat, Raised, Sunken }
    public class ColorControl : Control
    {
        private CustomBorderStyle _borderStyle = CustomBorderStyle.None;
        private ToolTip _toolTip;
        private.IContainer components;

        [DefaultValue(CustomBorderStyle.None)]
        public CustomBorderStyle BorderStyle
        {
            get { return _borderStyle; }
            set
            {
                _borderStyle = value;
                Refresh();
            }
        }

        public ColorControl()
        {
            InitializeComponent();

            SetStyle(ControlStyles.AllPaintingInWmPaint, true);
            SetStyle(ControlStyles.UserPaint, true);
            SetStyle(ControlStyles.OptimizedDoubleBuffer, true);
            UpdateStyles();

            AutoSize = false;
        }

        protected override void OnPaint(PaintEventArgs e)
        {
            base.OnPaint(e);

            e.Graphics.FillRectangle(new SolidBrush(ForeColor), ClientRectangle);

            if (BorderStyle == CustomBorderStyle.Dotted)
            {
                ControlPaint.DrawBorder(e.Graphics, ClientRectangle, SystemColors.WindowFrame, 1, SystemPens.DottedLine);
            }
            else if (BorderStyle == CustomBorderStyle.Dashed)
            {
                ControlPaint.DrawBorder(e.Graphics, ClientRectangle, SystemColors.WindowFrame, 1, SystemPens.DashDotDotLine);
            }
        }
    }
}
```

```
        }

        else if (BorderStyle != CustomBorderStyle.None)
        {
            Border3DStyle style;

            switch (BorderStyle)
            {
                case CustomBorderStyle.Bump: style = Border3DS...
                case CustomBorderStyle.Etched: style = Border3D...
                case CustomBorderStyle.Raised: style = Border3D...
                case CustomBorderStyle.Sunken: style = Border3D...
                default: style = Border3DStyle.Flat; break;
            }

            ControlPaint.DrawBorder3D(e.Graphics, ClientRectangle,
        }

        if (Focused)
        {
            Rectangle focusRect = new Rectangle(ClientRectangle.X - ...
            ControlPaint.DrawFocusRectangle(e.Graphics, focusRect)
        }
    }

protected override void OnGotFocus(EventArgs e)
{
    base.OnGotFocus(e);

    Invalidate();
}

protected override void OnLostFocus(EventArgs e)
{
    base.OnLostFocus(e);

    Invalidate();
}

protected override void OnMouseClick(MouseEventArgs e)
{
    base.OnMouseClick(e);

    Focus();

    if (e.Button == MouseButtons.Left)
    {
        using (ColorDialog fd = new ColorDialog())
        {
            fd.AnyColor = true;
            fd.FullOpen = true;
            fd.Color = Color.FromArgb(255, ForeColor);

            int[] customColors = Program.Cfg.CustomColors;

            if (customColors != null && customColors.Length > 0)
            {
                fd.CustomColors = customColors;
            }

            if (fd.ShowDialog() == DialogResult.OK)
            {
                ForeColor = fd.Color;
            }
        }
    }
}
```

```

        }
        ForeColor = fd.Color;
        Program.Cfg.CustomColors = fd.CustomCo
    }
}
else if (e.Button == MouseButtons.Right)
{
    ContextMenu menu = new ContextMenu();
    menu.MenuItems.Add(new MenuItem("Select Transparent Co
        menu.Show(this, e.Location);
    }
}

void menu_Click(object sender, EventArgs e)
{
    ForeColor = Color.Transparent;
}

protected override void OnKeyDown(KeyEventArgs e)
{
    using (ColorDialog fd = new ColorDialog())
    {
        fd.AnyColor = true;
        fd.FullOpen = true;
        fd.Color = ForeColor;

        int[] customColors = Program.Cfg.CustomColors;

        if (customColors != null && customColors.Length > 0)
        {
            fd.CustomColors = customColors;
        }

        if (fd.ShowDialog() == DialogResult.OK)
        {
            ForeColor = fd.Color;
            Program.Cfg.CustomColors = fd.CustomColors;
        }
    }

    base.OnKeyDown(e);
}

protected override void OnForeColorChanged(EventArgs e)
{
    String colorName = String.Empty, rgb, tooltip;

    if (ForeColor.ToKnownColor() != 0)
    {
        colorName = ForeColor.ToKnownColor().ToString();
    }
    else if (ForeColor.IsNamedColor)
    {
        colorName = ForeColor.ToString();
    }

    rgb = String.Format("R={0}; G={1}; B={2}", ForeColor.R, ForeCo
}

```

```

        if (colorName.Length > 0)
    {
        tooltip = String.Format("{0} ({1})", colorName, rgb);
    }
    else
    {
        tooltip = rgb;
    }

    _toolTip.SetToolTip(this, tooltip);

    base.OnForeColorChanged(e);
}

private void InitializeComponent()
{
    components = new Container();
    _toolTip = new ToolTip(components);
    SuspendLayout();
    //
    // ColorControl
    //
    _toolTip.SetToolTip(this, "Color");
    ResumeLayout(false);
}

}
}

```

Config.cs

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using System.ComponentModel;
using System.Diagnostics;
using System.Globalization;
using Microsoft.Win32;
using BytescoutScreenCapturingLib;

// NOTE: if you are getting error like "invalid image" related to loading the SDK's dll
// try to do the following:
// 1) remove the reference to the SDK by View - Solution Explorer
// then click on References, select Bytescout... reference name and right-click it and
// 2) To re-add click on the menu: Project - Add Reference
// 3) In "Add Reference" dialog switch to "COM" tab and find Bytescout...
// 4) Select it and click "Add"
// 5) Recompile the application
// Note: if you need to run on both x64 and x86 then please make sure you have set "Em

```

```

namespace ScreenCapturing
{
    public class Config
    {
        public int[] CustomColors { get { return Get<int[]>("CustomColors", null); } }
        public Point WindowLocation { get { return Get<Point>("WindowLocation", new Point(0, 0)); } }
        public string AudioDevice { get { return Get<string>("AudioDevice", ""); } }
        public string AudioLine { get { return Get<string>("AudioLine", ""); } }
        public string AviAudioCodec { get { return Get<string>("AviAudioCodec", "Wmaudio"); } }
        public string AviVideoCodec { get { return Get<string>("AviVideoCodec", "Wmvvideo"); } }
        public string WmvAudioCodec { get { return Get<string>("WmvAudioCodec", "Wmaudio"); } }
        public int WmvAudioFormat { get { return Get<int>("WmvAudioFormat", 28); } }
        public string WmvVideoCodec { get { return Get<string>("WmvVideoCodec", "Wmvvideo"); } }

        public bool EnableAudio { get { return Get<bool>("EnableAudio", false); } set { Options["EnableAudio"] = value; } }
        public bool ResizeOutputVideo { get { return Get<bool>("ResizeOutputVideo", true); } set { Options["ResizeOutputVideo"] = value; } }
        public int OutputWidth { get { return Get<int>("OutputWidth", 1024); } set { Options["OutputWidth"] = value; } }
        public int OutputHeight { get { return Get<int>("OutputHeight", 600); } set { Options["OutputHeight"] = value; } }
        public bool KeepAspectRatio { get { return Get<bool>("KeepAspectRatio", true); } set { Options["KeepAspectRatio"] = value; } }
        public float FPS { get { return Get<float>("FPS", 14.985f); } set { Options["FPS"] = value; } }
        public bool WriteLog { get { return Get<bool>("WriteLog", false); } set { Options["WriteLog"] = value; } }
        public bool DoNotShowMessage1 { get { return Get<bool>("DoNotShowMessage1", false); } set { Options["DoNotShowMessage1"] = value; } }
        public string LastUsedFolder { get { return Get<string>("LastUsedFolder", ""); } }

        public int SelectedVideoCodecTab { get { return Get<int>("SelectedVideoCodecTab", 0); } }
        public string WebCameraDevice { get { return Get<string>("WebCameraDevice", "Default"); } }
        public int WebCameraWindowX { get { return Get<int>("WebCameraWindowX", 0); } }
        public int WebCameraWindowY { get { return Get<int>("WebCameraWindowY", 0); } }
        public int WebCameraWindowWidth { get { return Get<int>("WebCameraWindowWidth", 640); } }
        public int WebCameraWindowHeight { get { return Get<int>("WebCameraWindowHeight", 480); } }
        public bool ShowMouseHotSpot { get { return Get<bool>("ShowMouseHotSpot", true); } }
        public bool CaptureMouseCursor { get { return Get<bool>("CaptureMouseCursor", true); } }
        public bool AnimateMouseClicks { get { return Get<bool>("AnimateMouseClicks", true); } }
        public bool AnimateMouseButtons { get { return Get<bool>("AnimateMouseButtons", true); } }
        public int MouseAnimationDuration { get { return Get<int>("MouseAnimationDuration", 100); } }
        public int MouseSpotRadius { get { return Get<int>("MouseSpotRadius", 10); } }
        public Color MouseHotSpotColor { get { return Get<Color>("MouseHotSpotColor", Color.White); } }
        public Color MouseCursorLeftClickAnimationColor { get { return Get<Color>("MouseCursorLeftClickAnimationColor", Color.White); } }
        public Color MouseCursorRightClickAnimationColor { get { return Get<Color>("MouseCursorRightClickAnimationColor", Color.White); } }
        public CaptureAreaBorderType CaptureAreaBorderType { get { return Get<CaptureAreaBorderType>("CaptureAreaBorderType", CaptureAreaBorderType.Solid); } }
        public Color CaptureAreaBorderColor { get { return Get<Color>("CaptureAreaBorderColor", Color.Black); } }
        public int CaptureAreaBorderWidth { get { return Get<int>("CaptureAreaBorderWidth", 1); } }

        private readonly String _strKey;
        public Dictionary<string, object> Options = new Dictionary<string, object>();
        public bool Subsection;

        public Config()
        {
            _strKey = Program.RegistryKey;
        }

        public Config(String section)
        {
            Debug.Assert(!String.IsNullOrEmpty(section));

            Subsection = true;
            _strKey = Program.RegistryKey + "\\\" + section;
        }
    }
}

```

```
public T Get<T>(String optionName, Object defaultValue)
{
    Debug.Assert(!String.IsNullOrEmpty(optionName));

    object value;

    if (Options.TryGetValue(optionName, out value))
    {
        return (T) value;
    }
    else
    {
        String s = (String) Registry.GetValue("HKEY_CURRENT_USI
        if (s == null)
        {
            return (T) defaultValue;
        }
        else
        {
            TypeConverter tc;

            if (typeof(T) == typeof(int[]))
            {
                tc = new IntegerArrayConverter();
            }
            else
            {
                tc = TypeDescriptor.GetConverter(typeof(T));
            }

            try
            {
                value = tc.ConvertFromString(s);
            }
            catch (Exception)
            {
                value = defaultValue;
            }
        }

        Options.Add(optionName, value);
    }
    return (T) value;
}
}

public void Set(String optionName, Object value)
{
    Debug.Assert(!String.IsNullOrEmpty(optionName));

    Options[optionName] = value;
}

public void Save()
{
    RegistryKey key = Registry.CurrentUser.CreateSubKey(_strKey);

    if (key != null)
    {
```

```

        foreach (KeyValuePair<string, object> de in Options)
    {
        TypeConverter tc;
        String s;

        if (de.Value == null)
        {
            s = "";
        }
        else
        {
            if (de.Value is int[])
            {
                tc = new IntegerArrayConverter();
            }
            else
            {
                tc = TypeDescriptor.GetConverter(de.Value);
            }

            s = tc.ConvertToString(de.Value);
        }

        key.SetValue(de.Key, s);
    }

    key.Flush();
}
}

public class IntegerArrayConverter : TypeConverter
{
    public override bool CanConvertFrom(ITypeDescriptorContext context, Type sourceType)
    {
        if (sourceType == typeof(string))
        {
            return true;
        }

        return base.CanConvertFrom(context, sourceType);
    }

    public override object ConvertFrom(ITypeDescriptorContext context, CultureInfo culture, string value)
    {
        if (value is string)
        {
            string[] ss = ((string) value).Split(new char[] { ';' });
            List<int> list = new List<int>();

            foreach (string s in ss)
            {
                if (!String.IsNullOrEmpty(s))
                {
                    list.Add(Int32.Parse(s));
                }
            }

            return list.ToArray();
        }
    }
}

```

```

        return base.ConvertFrom(context, culture, value);
    }

    public override bool CanConvertTo(ITypeDescriptorContext context, Type
    {
        if (destinationType == typeof(string))
        {
            return true;
        }

        return base.CanConvertTo(context, destinationType);
    }

    public override object ConvertTo(ITypeDescriptorContext context, Cultur
    {
        if (destinationType == typeof(string) && value is int[])
        {
            StringBuilder sb = new StringBuilder();

            foreach (int x in (int[]) value)
            {
                if (sb.Length > 0)
                {
                    sb.Append(";");
                }

                sb.Append(x.ToString());
            }

            return sb.ToString();
        }

        return base.ConvertTo(context, culture, value, destinationType);
    }
}
}

```

CustomMessageBox.cs

```

using System;
using System.Windows.Forms;

namespace ScreenCapturing
{
    public partial class CustomMessageBox : Form
    {
        public bool DoNotShow = false;

        public CustomMessageBox(string title, string text)
        {
            this.Text = title;
            this.Content = text;
        }
    }
}

```

```
{  
    InitializeComponent();  
  
    base.Text = title;  
    label1.Text = text;  
}  
  
private void cbDoNotAsk_CheckedChanged(object sender, EventArgs e)  
{  
    DoNotShow = cbDoNotShow.Checked;  
}  
}  
}
```

CustomMessageBox.designer.cs

```
namespace ScreenCapturing  
{  
    partial class CustomMessageBox  
    {  
        /// <summary>  
        /// Required designer variable.  
        /// </summary>  
        private System.ComponentModel.IContainer components = null;  
  
        /// <summary>  
        /// Clean up any resources being used.  
        /// </summary>  
        /// <param name="disposing">true if managed resources should be disposed.  
        protected override void Dispose(bool disposing)  
        {  
            if (disposing && (components != null))  
            {  
                components.Dispose();  
            }  
            base.Dispose(disposing);  
        }  
  
        #region Windows Form Designer generated code  
  
        /// <summary>  
        /// Required method for Designer support - do not modify  
        /// the contents of this method with the code editor.  
        /// </summary>  
        private void InitializeComponent()  
        {  
            this.label1 = new System.Windows.Forms.Label();  
            this.btnExit = new System.Windows.Forms.Button();  
            this.cbDoNotShow = new System.Windows.Forms.CheckBox();  
            this.SuspendLayout();  
        }  
    }  
}
```

```
//  
// label1  
//  
this.label1.Location = new System.Drawing.Point(12, 9);  
this.label1.Name = "label1";  
this.label1.Size = new System.Drawing.Size(335, 34);  
this.label1.TabIndex = 0;  
this.label1.Text = "label1";  
this.label1.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;  
//  
// btnContinue  
//  
this.btnContinue.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));  
this.btnContinue.DialogResult = System.Windows.Forms.DialogResult.OK;  
this.btnContinue.Location = new System.Drawing.Point(272, 82);  
this.btnContinue.Name = "btnContinue";  
this.btnContinue.Size = new System.Drawing.Size(75, 23);  
this.btnContinue.TabIndex = 0;  
this.btnContinue.Text = "&Continue";  
this.btnContinue.UseVisualStyleBackColor = true;  
//  
// cbDoNotShow  
//  
this.cbDoNotShow.AutoSize = true;  
this.cbDoNotShow.Location = new System.Drawing.Point(12, 86);  
this.cbDoNotShow.Name = "cbDoNotShow";  
this.cbDoNotShow.Size = new System.Drawing.Size(129, 17);  
this.cbDoNotShow.TabIndex = 2;  
this.cbDoNotShow.Text = "Do not show anymore";  
this.cbDoNotShow.UseVisualStyleBackColor = true;  
this.cbDoNotShow.CheckedChanged += new System.EventHandler(this.cbDoNotAsk.  
//  
// CustomMessageBox  
//  
this.AcceptButton = this.btnContinue;  
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);  
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;  
this.AutoSize = true;  
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.GrowAndShrink;  
this.ClientSize = new System.Drawing.Size(359, 117);  
this.Controls.Add(this.cbDoNotShow);  
this.Controls.Add(this.btnContinue);  
this.Controls.Add(this.label1);  
this.MaximizeBox = false;  
this.MinimizeBox = false;  
this.Name = "CustomMessageBox";  
this.ShowIcon = false;  
this.ShowInTaskbar = false;  
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;  
this.Text = "Information";  
this.ResumeLayout(false);  
this.PerformLayout();  
this.PerformLayout();  
  
}  
  
#endregion  
  
private System.Windows.Forms.Label label1;  
private System.Windows.Forms.Button btnContinue;  
private System.Windows.Forms.CheckBox cbDoNotShow;
```

```
    }  
}
```

MainForm.Designer.cs

```
namespace ScreenCapturing  
{  
    sealed partial class MainForm  
    {  
        /// <summary>  
        /// Required designer variable.  
        /// </summary>  
        private System.ComponentModel.IContainer components = null;  
  
        /// <summary>  
        /// Clean up any resources being used.  
        /// </summary>  
        /// <param name="disposing">true if managed resources should be disposed.  
        protected override void Dispose(bool disposing)  
        {  
            if (disposing && (components != null))  
            {  
                components.Dispose();  
            }  
            base.Dispose(disposing);  
        }  
  
        #region Windows Form Designer generated code  
  
        /// <summary>  
        /// Required method for Designer support - do not modify  
        /// the contents of this method with the code editor.  
        /// </summary>  
        private void InitializeComponent()  
        {  
            System.ComponentModel.ComponentResourceManager resources = new  
            this.statusStrip = new System.Windows.Forms.StatusStrip();  
            this.label1 = new System.Windows.Forms.ToolStripStatusLabel();  
            this.toolStrip1 = new System.Windows.Forms.ToolStrip();  
            this.tsbRecord = new System.Windows.Forms.ToolStripDropDownButton();  
            this.tsmiRectangularRegion2 = new System.Windows.Forms.ToolStripItem();  
            this.tsmiMouseRegion2 = new System.Windows.Forms.ToolStripMenuItem();  
            this.tsmiWindow2 = new System.Windows.Forms.ToolStripMenuItem();  
            this.tsmiFullScreen2 = new System.Windows.Forms.ToolStripMenuItem();  
            this.tsbRecordWithCam = new System.Windows.Forms.ToolStripDropDownButton();  
            this.tsmiRectangularRegion = new System.Windows.Forms.ToolStripItem();  
            this.tsmiMouseRegion = new System.Windows.Forms.ToolStripMenuItem();  
            this.tsmiWindow = new System.Windows.Forms.ToolStripMenuItem();  
            this.tsmiFullScreen = new System.Windows.Forms.ToolStripMenuItem();  
            this.toolStripSeparator2 = new System.Windows.Forms.ToolStripSeparator();  
        }  
    }  
}
```

```
this.tsbSettings = new System.Windows.Forms.ToolStripButton();
this.tsbPlay = new System.Windows.Forms.ToolStripButton();
this.toolStripSeparator1 = new System.Windows.Forms.ToolStripSeparator();
this.statusStrip.SuspendLayout();
this.toolStrip1.SuspendLayout();
this.SuspendLayout();
//
// statusBar
//
this.statusBar.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
    this.label1});
this.statusBar.Location = new System.Drawing.Point(0, 85);
this.statusBar.Name = "statusBar";
this.statusBar.Size = new System.Drawing.Size(417, 22);
this.statusBar.SizingGrip = false;
this.statusBar.TabIndex = 1;
this.statusStrip.Text = "statusStrip1";
//
// label1
//
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(402, 17);
this.label1.Spring = true;
this.label1.Text = "Ready";
this.label1.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// toolStrip1
//
this.toolStrip1.BackgroundImage = global::ScreenCapturing.Properties.Resources.ToolStripBackground;
this.toolStrip1.GripStyle = System.Windows.Forms.ToolStripGripStyle.Hidden;
this.toolStrip1.ImageScalingSize = new System.Drawing.Size(64, 64);
this.toolStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
    this.tsbRecord,
    this.tsbRecordWithCam,
    this.toolStripSeparator2,
    this.tsbPlay,
    this.toolStripSeparator1,
    this.tsbSettings});
this.toolStrip1.Location = new System.Drawing.Point(0, 0);
this.toolStrip1.Name = "toolStrip1";
this.toolStrip1.Size = new System.Drawing.Size(417, 86);
this.toolStrip1.TabIndex = 0;
this.toolStrip1.Text = "toolStrip1";
//
// tsbRecord
//
this.tsbRecord.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
    this.tsmiRectangularRegion2,
    this.tsmiMouseRegion2,
    this.tsmiWindow2,
    this.tsmiFullScreen2});
this.tsbRecord.Image = global::ScreenCapturing.Properties.Resources.RecordIcon;
this.tsbRecord.ImageTransparentColor = System.Drawing.Color.Magenta;
this.tsbRecord.Margin = new System.Windows.Forms.Padding(20, 10, 0, 0);
this.tsbRecord.Name = "tsbRecord";
this.tsbRecord.Size = new System.Drawing.Size(77, 83);
this.tsbRecord.Text = "Record";
this.tsbRecord.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageBeforeText;
this.tsbRecord.ToolTipText = "Records a movie using the current video source";
this.tsbRecord.Click += new System.EventHandler(this.tsbRecord_Click);
```

```
//  
// tsmiRectangularRegion2  
//  
this.tsmiRectangularRegion2.Name = "tsmiRectangularRegion2";  
this.tsmiRectangularRegion2.Size = new System.Drawing.Size(211,  
this.tsmiRectangularRegion2.Text = "Rectangular region";  
this.tsmiRectangularRegion2.Click += new System.EventHandler(this.tsmiRectangularRegion2_Click);  
//  
// tsmiMouseRegion2  
//  
this.tsmiMouseRegion2.Name = "tsmiMouseRegion2";  
this.tsmiMouseRegion2.Size = new System.Drawing.Size(211, 22);  
this.tsmiMouseRegion2.Text = "Region around the Mouse";  
this.tsmiMouseRegion2.Click += new System.EventHandler(this.tsmiMouseRegion2_Click);  
//  
// tsmiWindow2  
//  
this.tsmiWindow2.Name = "tsmiWindow2";  
this.tsmiWindow2.Size = new System.Drawing.Size(211, 22);  
this.tsmiWindow2.Text = "Window";  
this.tsmiWindow2.Visible = false;  
//  
// tsmiFullScreen2  
//  
this.tsmiFullScreen2.Name = "tsmiFullScreen2";  
this.tsmiFullScreen2.Size = new System.Drawing.Size(211, 22);  
this.tsmiFullScreen2.Text = "Full Screen";  
this.tsmiFullScreen2.Click += new System.EventHandler(this.tsmiFullScreen2_Click);  
//  
// tsbRecordWithCam  
//  
this.tsbRecordWithCam.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {  
    this.tsmiRectangularRegion,  
    this.tsmiMouseRegion,  
    this.tsmiWindow,  
    this.tsmiFullScreen});  
    this.tsbRecordWithCam.Image = global::ScreenCapturing.Properties.Resources.Record;  
    this.tsbRecordWithCam.ImageTransparentColor = System.Drawing.Color.White;  
    this.tsbRecordWithCam.Margin = new System.Windows.Forms.Padding(0);  
    this.tsbRecordWithCam.Name = "tsbRecordWithCam";  
    this.tsbRecordWithCam.Size = new System.Drawing.Size(131, 83);  
    this.tsbRecordWithCam.Text = "Record with webcam";  
    this.tsbRecordWithCam.TextImageRelation = System.Windows.Forms.TextImageRelation.TextBeforeImage;  
    this.tsbRecordWithCam.ToolTipText = "Records a movie using the camera";  
    this.tsbRecordWithCam.Click += new System.EventHandler(this.tsbRecordWithCam_Click);  
//  
// tsmiRectangularRegion  
//  
this.tsmiRectangularRegion.Name = "tsmiRectangularRegion";  
this.tsmiRectangularRegion.Size = new System.Drawing.Size(211,  
this.tsmiRectangularRegion.Text = "Rectangular region";  
this.tsmiRectangularRegion.Click += new System.EventHandler(this.tsmiRectangularRegion_Click);  
//  
// tsmiMouseRegion  
//  
this.tsmiMouseRegion.Name = "tsmiMouseRegion";  
this.tsmiMouseRegion.Size = new System.Drawing.Size(211, 22);  
this.tsmiMouseRegion.Text = "Region around the Mouse";  
this.tsmiMouseRegion.Click += new System.EventHandler(this.tsmiMouseRegion_Click);  
//
```

```
// tsmiWindow
//
this.tsmiWindow.Name = "tsmiWindow";
this.tsmiWindow.Size = new System.Drawing.Size(211, 22);
this.tsmiWindow.Text = "Window";
this.tsmiWindow.Visible = false;
//
// tsmiFullScreen
//
this.tsmiFullScreen.Name = "tsmiFullScreen";
this.tsmiFullScreen.Size = new System.Drawing.Size(211, 22);
this.tsmiFullScreen.Text = "Full Screen";
this.tsmiFullScreen.Click += new System.EventHandler(this.tsmiF
//
// toolStripSeparator2
//
this.toolStripSeparator2.Name = "toolStripSeparator2";
this.toolStripSeparator2.Size = new System.Drawing.Size(6, 86)
//
// tsbSettings
//
this.tsbSettings.Image = global::ScreenCapturing.Properties.Res
this.tsbSettings.ImageScaling = System.Windows.Forms.ToolStripItem
this.tsbSettings.ImageTransparentColor = System.Drawing.Color.M
this.tsbSettings.Name = "tsbSettings";
this.tsbSettings.Size = new System.Drawing.Size(68, 83);
this.tsbSettings.Text = "Settings";
this.tsbSettings.TextImageRelation = System.Windows.Forms.TextImage
this.tsbSettings.ToolTipText = "Shows settings dialog";
this.tsbSettings.Click += new System.EventHandler(this.tsbSetting
//
// tsbPlay
//
this.tsbPlay.Enabled = false;
this.tsbPlay.Image = global::ScreenCapturing.Properties.Resource
this.tsbPlay.ImageTransparentColor = System.Drawing.Color.Magenta
this.tsbPlay.Name = "tsbPlay";
this.tsbPlay.Size = new System.Drawing.Size(68, 83);
this.tsbPlay.Text = "Play";
this.tsbPlay.TextImageRelation = System.Windows.Forms.TextImageR
this.tsbPlay.ToolTipText = "Tries to play the recorded movie in
this.tsbPlay.Click += new System.EventHandler(this.tsbPlay_Click
//
// toolStripSeparator1
//
this.toolStripSeparator1.Name = "toolStripSeparator1";
this.toolStripSeparator1.Size = new System.Drawing.Size(6, 86)
//
// MainForm
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(417, 107);
this.Controls.Add(this.statusStrip);
this.Controls.Add(this.toolStrip1);
this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;
this.Icon = ((System.Drawing.Icon) (resources.GetObject("$this
this.MaximizeBox = false;
this.Name = "MainForm";
this.StartPosition = System.Windows.Forms.FormStartPosition.Man
```

```

        this.Text = "Bytescout Screen Capturing";
        this.statusStrip.ResumeLayout(false);
        this.statusStrip.PerformLayout();
        this.toolStrip1.ResumeLayout(false);
        this.toolStrip1.PerformLayout();
        this.ResumeLayout(false);
        this.PerformLayout();

    }

#endregion

    private System.Windows.Forms.ToolStrip toolStrip1;
    private System.Windows.Forms.ToolStripButton tsbSettings;
    private System.Windows.Forms.ToolStripButton tsbPlay;
    private System.Windows.Forms.StatusStrip statusStrip;
    private System.Windows.Forms.ToolStripDropDownButton tsbRecordWithCam;
    private System.Windows.Forms.ToolStripItem tsmiRectangularRegion;
    private System.Windows.Forms.ToolStripItem tsmiWindow;
    private System.Windows.Forms.ToolStripItem tsmiFullScreen;
    private System.Windows.Forms.ToolStripItem tsmiMouseRegion;
    private System.Windows.Forms.ToolStripStatusLabel label1;
    private System.Windows.Forms.ToolStripDropDownButton tsbRecord;
    private System.Windows.Forms.ToolStripItem tsmiRectangularRegion2;
    private System.Windows.Forms.ToolStripItem tsmiMouseRegion2;
    private System.Windows.Forms.ToolStripItem tsmiWindow2;
    private System.Windows.Forms.ToolStripItem tsmiFullScreen2;
    private System.Windows.Forms.ToolStripSeparator toolStripSeparator1;
    private System.Windows.Forms.ToolStripSeparator toolStripSeparator2;
}

}

```

MainForm.cs

```

using System;
using System.ComponentModel;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Runtime.InteropServices;
using System.Threading;
using System.Windows.Forms;
using ScreenCapturing.Properties;
using BytescoutScreenCapturingLib;

// NOTE: if you are getting error like "invalid image" related to loading the SDK's dl
// try to do the following:
// 1) remove the reference to the SDK by View - Solution Explorer

```

```

// then click on References, select Bytescout... reference name and right-click it and
// 2) To re-add click on the menu: Project - Add Reference
// 3) In "Add Reference" dialog switch to "COM" tab and find Bytescout...
// 4) Select it and click "Add"
// 5) Recompile the application
// Note: if you need to run on both x64 and x86 then please make sure you have set "Embe

namespace ScreenCapturing
{
    public sealed partial class MainForm : Form
    {
        private bool _recording;
        private ToolStripDropDown _dropDownItems;
        private Rectangle _recordingRegion = Rectangle.Empty;
        private Thread _capturingThread;
        private CapturingThreadData _capturingThreadData;
        private string _lastRecordedFile;

        private const int MYKEYID = 0;
        private const int WM_HOTKEY = 0x0312;
        private const int WM_MOUSEMOVE = 0x0200;
        private const int MOD_ALT = 1;
        private const int MOD_CONTROL = 2;
        private const int MOD_SHIFT = 4;

        [DllImport("user32.dll")]
        private static extern int RegisterHotKey(IntPtr hWnd, int id, int modifier, Key
        [DllImport("user32.dll")]
        private static extern bool UnregisterHotKey(IntPtr hWnd, int id);
        [DllImport("user32.dll")]
        private static extern bool SetForegroundWindow(IntPtr hWnd);

        public MainForm()
        {
            InitializeComponent();

            Point location = Program.Cfg.WindowLocation;

            if (location.IsEmpty)
            {
                StartPosition = FormStartPosition.CenterScreen;
            }
            else
            {
                Location = location;
            }

            Text = Application.ProductName + " " + Application.ProductVers
        }

        private void tsmiFullScreen_Click(object sender, EventArgs e)
        {
            _recordingRegion = Screen.PrimaryScreen.Bounds;

            StartRecording(CaptureAreaType.catScreen, sender == tsmiFullSc
        }

        private void tsmiMouseRegion_Click(object sender, EventArgs e)
        {

```

```
        _recordingRegion.Width = Program.Cfg.OutputWidth;
        _recordingRegion.Height = Program.Cfg.OutputHeight;

    StartRecording(CaptureAreaType.catMouse, sender == tsmiMouseRegion);
}

private void tsbRecord_Click(object sender, EventArgs e)
{
    if (_recording)
    {
        StopRecording();
    }
}

private void tsbPlay_Click(object sender, EventArgs e)
{
    Process prc = new Process();
    prc.StartInfo.FileName = _lastRecordedFile;
    prc.StartInfo.UseShellExecute = true;

    try
    {
        prc.Start();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, Application.ProductName);
    }
}

private void tsbSettings_Click(object sender, EventArgs e)
{
    using (SettingsForm f = new SettingsForm())
    {
        f.ShowDialog();
    }
}

private void StartRecording(CaptureAreaType captureType, bool withCam)
{
    _recording = true;

    if (withCam)
    {
        _dropDownItems = tsbRecordWithCam.DropDown;
        tsbRecord.Enabled = false;
        tsbRecordWithCam.ShowDropDownArrow = false;
        tsbRecordWithCam.DropDown = null;
        tsbRecordWithCam.Image = Resources.stop;
        tsbRecordWithCam.Text = "Stop";
    }
    else
    {
        _dropDownItems = tsbRecord.DropDown;
        tsbRecordWithCam.Enabled = false;
        tsbRecord.ShowDropDownArrow = false;
        tsbRecord.DropDown = null;
        tsbRecord.Image = Resources.stop;
        tsbRecord.Text = "Stop";
    }
}
```

```
        statusBar.Items[0].Text = "Recording started";
        tsbPlay.Enabled = false;

        if (!Program.Cfg.DoNotShowMessage1)
        {
            CustomMessageBox mb = new CustomMessageBox(Application
mb.StartPosition = FormStartPosition.CenterScreen;
            mb.ShowDialog();
            Program.Cfg.DoNotShowMessage1 = mb.DoNotShow;
        }

        RegisterHotKey(Handle, MYKEYID, MOD_CONTROL + MOD_SHIFT, Keys.F12);

        _capturingThreadData = new CapturingThreadData();

        _capturingThreadData.CaptureType = captureType;
        _capturingThreadData.CaptureRectangle = _recordingRegion;
        _capturingThreadData.ShowWebCamStream = withCam;

        _capturingThread = new Thread(CapturingThread.ThreadProc);
        _capturingThread.Start(_capturingThreadData);

        WindowState = FormWindowState.Minimized;
    }

protected override void WndProc(ref Message m)
{
    if (m.Msg == WM_HOTKEY && m.WParam.ToInt32() == MYKEYID)
    {
        if (_recording)
        {
            StopRecording();
        }
    }

    base.WndProc(ref m);
}

private void StopRecording()
{
    Cursor = Cursors.WaitCursor;

    try
    {
        _capturingThread.Interrupt();
        _capturingThread.Join();
    }
    finally
    {
        Cursor = Cursors.Default;
    }

    _recording = false;

    if (!tsbRecord.Enabled)
    {
        tsbRecordWithCam.ShowDropDownArrow = true;
        tsbRecordWithCam.DropDown = _dropDownItems;
        tsbRecordWithCam.Image = Resources.record_cam;
    }
}
```

```

        tsbRecordWithCam.Text = "Record with webcam";
        _dropDownItems = null;
        tsbRecord.Enabled = true;
    }
    else
    {
        tsbRecord.ShowDropDownArrow = true;
        tsbRecord.DropDown = _dropDownItems;
        tsbRecord.Image = Resources.record;
        tsbRecord.Text = "Record";
        _dropDownItems = null;
        tsbRecordWithCam.Enabled = true;
    }

    UnregisterHotKey(Handle, MYKEYID);
    WindowState = FormWindowState.Normal;
    SetForegroundWindow(Handle);

    if (_capturingThreadData.Result != 0)
    {
        statusBar.Items[0].Text = "Recording failed";
        MessageBox.Show("Capturing failed.\n\nError: " + _capturingThreadData.Error);
    }
    else
    {
        SaveFileDialog dlg = new SaveFileDialog();

        if (Program.Cfg.SelectedVideoCodecTab == 0)
        {
            dlg.DefaultExt = "*.wmv";
            dlg.Filter = "WMV files (*.wmv)|*.wmv|All files (*.*)|*.*";
        }
        else
        {
            dlg.DefaultExt = "*.avi";
            dlg.Filter = "AVI files (*.avi)|*.avi|All files (*.*)|*.*";
        }

        DateTime now = DateTime.Now;
        string fileName = string.Format("Screencast-{0}-{1:00}-{2:00}-{3:00}.{4}", now.ToString("yy"), now.ToString("MM"), now.ToString("ss"), now.ToString("fff"), Program.Cfg.VideoCodec);
        dlg.FileName = fileName;
        dlg.Title = "Save captured video as";
        dlg.InitialDirectory = Program.Cfg.LastUsedFolder;

        if (dlg.ShowDialog() == DialogResult.OK)
        {
            File.Copy(_capturingThreadData.TempFile, dlg.FileName, true);

            _lastRecordedFile = dlg.FileName;
            statusBar.Items[0].Text = "Successfully recorded \\" + Path.GetFileName(_lastRecordedFile);
            tsbPlay.Enabled = File.Exists(_lastRecordedFile);
            Program.Cfg.LastUsedFolder = Path.GetDirectoryName(dlg.FileName);

            // open saved video file in the default media player
            Process.Start(dlg.FileName);
        }
        else
        {
            statusBar.Items[0].Text = "Canceled";
        }
    }
}

```

```

        }

        try
        {
            File.Delete(_capturingThreadData.TempFile);
        }
        catch
        {
        }
    }

protected override void OnLocationChanged(EventArgs e)
{
    Program.Cfg.WindowLocation = Location;
    base.OnLocationChanged(e);
}

protected override void OnClosing(CancelEventArgs e)
{
    if (_recording)
    {
        StopRecording();
    }

    Program.Cfg.Save();
    base.OnClosing(e);
}

private void tsmiRectangularRegion_Click(object sender, EventArgs e)
{
    RegionSelector rs = new RegionSelector();

    if (rs.ShowDialog() != DialogResult.Abort)
    {
        _recordingRegion = rs.SelectedRectangle;
        StartRecording(CaptureAreaType.catRegion, sender == tsmiRectangularRegion);
    }
}
}

```

MouseHook.cs

```

using System;
using System.Runtime.InteropServices;
using System.Reflection;
using System.ComponentModel;
using System.Windows.Forms;

```

```
namespace ScreenCapturing
{
    [StructLayout(LayoutKind.Sequential)]
    public struct POINT
    {
        public int X;
        public int Y;

        public POINT(int x, int y)
        {
            X = x;
            Y = y;
        }

        public static implicit operator System.Drawing.Point(POINT p)
        {
            return new System.Drawing.Point(p.X, p.Y);
        }

        public static implicit operator POINT(System.Drawing.Point p)
        {
            return new POINT(p.X, p.Y);
        }
    }

    public class MouseHook
    {
        private delegate int HookProc(int nCode, int wParam, IntPtr lParam);
        private static HookProc MouseHookProcedure;
        public event MouseEventHandler MouseActivity;
        private int hMouseHook = 0;

        private const int WM_LBUTTONDOWN = 0x201;
        private const int WM_RBUTTONDOWN = 0x204;
        private const int WM_MOUSEWHEEL = 0x020A;
        private const int WM_LBUTTONDOWNDBLCLK = 0x203;
        private const int WM_RBUTTONDOWNDBLCLK = 0x206;

        private const int WH_MOUSE = 7;
        private const int WH_MOUSE_LL = 14;

        [DllImport("user32.dll", CharSet = CharSet.Auto, CallingConvention = CallingConvention.Winapi)]
        private static extern int SetWindowsHookEx(int idHook, HookProc lpfn, IntPtr hMod, int dwThreadId);

        [DllImport("user32.dll", CharSet = CharSet.Auto, CallingConvention = CallingConvention.Winapi)]
        private static extern int CallNextHookEx(int idHook, int nCode, int wParam, IntPtr lParam);

        [DllImport("user32.dll", CharSet = CharSet.Auto, CallingConvention = CallingConvention.Winapi)]
        private static extern int UnhookWindowsHookEx(int idHook);

        [StructLayout(LayoutKind.Sequential)]
        private class MouseLLHookStruct
        {
            public POINT pt;
            public int mouseData;
            public int flags;
            public int time;
            public int dwExtraInfo;
        }
    }
}
```

```
[StructLayout(LayoutKind.Sequential)]
private class MouseHookStruct
{
    public POINT pt;
    public int hwnd;
    public int wHitTestCode;
    public int dwExtraInfo;
}

public void SetHook()
{
    MouseHookProcedure = new HookProc(MouseHookProc);
    IntPtr modulehandle = Marshal.GetHINSTANCE(Assembly.GetExecutingAssembly().GetManifestResourceStream("MouseHook.dll"), 0);

    hMouseHook = SetWindowsHookEx(WH_MOUSE, MouseHookProcedure, modulehandle, 0);
}

if (hMouseHook == 0)
{
    int errorCode = Marshal.GetLastWin32Error();

    ReleaseHook(false);

    throw new Win32Exception(errorCode);
}
}

public void ReleaseHook(bool throwExceptions)
{
    if (hMouseHook != 0)
    {
        int retMouse = UnhookWindowsHookEx(hMouseHook);

        hMouseHook = 0;

        if (retMouse == 0 && throwExceptions)
        {
            throw new Win32Exception(Marshal.GetLastWin32Error());
        }
    }
}

private int MouseHookProc(int nCode, int wParam, IntPtr lParam)
{
    if ((nCode >= 0) && (MouseActivity != null))
    {
        MouseLLHookStruct mouseHookStruct =
            (MouseLLHookStruct) Marshal.PtrToStructure(lParam, typeof(MouseLLHookStruct));

        MouseButtons button = MouseButtons.None;
        short mouseDelta = 0;

        switch (wParam)
        {
            case WM_LBUTTONDOWN:
                button = MouseButtons.Left;
                break;
            case WM_RBUTTONDOWN:
                button = MouseButtons.Right;
                break;
            case WM_MOUSEWHEEL:

```

```

        mouseDelta = (short) ((mouseHookStruct.mouseData >> 16) & 0xffff);
    }

    int clickCount = 0;

    if (button != MouseButtons.None)
        if (wParam == WM_LBUTTONDOWNDBLCLK || wParam == WM_RBUTTONDOWNDBLCLK) clickCount = 2;
        else clickCount = 1;

    MouseEventArgs e = new MouseEventArgs(button, clickCount, mouseHookStruct.pt.X, mouseHookStruct.pt.Y, mouseDelta);

    MouseActivity(this, e);
}

return CallNextHookEx(hMouseHook, nCode, wParam, lParam);
}
}
}

```

Program.cs

```

using System;
using System.Windows.Forms;

namespace ScreenCapturing
{
    static class Program
    {
        public static Config Cfg = null;

        public static string RegistryKey = "Software\\Screen Capturing";
        public static string PostCommand = String.Empty;

        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);

            Cfg = new Config();

            MainForm mainForm = new MainForm();

            Application.Run(mainForm);
        }
    }
}

```

RegionSelector.cs

```
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Drawing.Imaging;
using System.IO;
using System.Windows.Forms;

namespace ScreenCapturing
{
    public sealed class RegionSelector : Form
    {
        private Bitmap _bmpScreenshot;
        private Rectangle _selectedRectangle = Rectangle.Empty;
        private bool _selecting = false;
        private readonly Cursor _cursor1;
        private readonly Cursor _cursor2;

        public RegionSelector()
        {
            _cursor1 = new Cursor(Path.Combine(Application.StartupPath, "cu
            _cursor2 = new Cursor(Path.Combine(Application.StartupPath, "cu

            Cursor = _cursor1;

            SetStyle(ControlStyles.AllPaintingInWmPaint | ControlStyles.UserPaint | Co
            ControlBox = false;
            FormBorderStyle = FormBorderStyle.None;
            Name = "Region Selector";
            StartPosition = FormStartPosition.Manual;
            Location = new Point(0, 0);
            Size = SystemInformation.PrimaryMonitorSize;
            TopMost = true;

            _bmpScreenshot = new Bitmap(Screen.PrimaryScreen.Bounds.Width,
                Screen.PrimaryScreen.Bounds.Height, PixelFormat.Format32bppArgb);

            using (Graphics g = Graphics.FromImage(_bmpScreenshot))
            {
                g.CopyFromScreen(Screen.PrimaryScreen.Bounds.X, Screen.PrimaryScreen.B
                    Screen.PrimaryScreen.Bounds.Size, CopyPixelOperation.SourceCopy);
            }
        }

        public Rectangle SelectedRectangle
        {
            get { return NormalizeRectangle(_selectedRectangle); }
            set { _selectedRectangle = value; }
        }
    }
}
```

```
protected override void OnPaint(PaintEventArgs e)
{
    e.Graphics.DrawImage(_bmpScreenshot, new Rectangle(0, 0, _bmpScreenshot.Width, _bmpScreenshot.Height));

    Region region = new Region(new Rectangle(0, 0, _bmpScreenshot.Width, _bmpScreenshot.Height));
    Rectangle rectangle = SelectedRectangle;

    region.Exclude(rectangle);

    using (Brush brush = new SolidBrush(Color.FromArgb(174, Color.White)))
    {
        e.Graphics.FillRegion(brush, region);
    }

    if (!rectangle.IsEmpty)
    {
        using (Pen pen = new Pen(Color.Red, 1))
        {
            pen.Alignment = PenAlignment.Outset;
            e.Graphics.DrawRectangle(pen, rectangle);
        }
    }
}

public static Rectangle NormalizeRectangle(Rectangle r)
{
    Rectangle result = r;

    if (result.Left > result.Right)
    {
        int width = result.Left - result.Right;
        result.X = result.Right;
        result.Width = System.Math.Abs(width);
    }

    if (result.Top > result.Bottom)
    {
        int height = result.Top - result.Bottom;
        result.Y = result.Bottom;
        result.Height = System.Math.Abs(height);
    }

    return result;
}

protected override void OnMouseMove(MouseEventArgs e)
{
    if (_selecting)
    {
        _selectedRectangle.Width = e.X - _selectedRectangle.X;
        _selectedRectangle.Height = e.Y - _selectedRectangle.Y;

        Invalidate();
    }
}

protected override void OnMouseUp(MouseEventArgs e)
{
    if (!_selecting)
```

```

        {
            _selecting = true;
            _selectedRectangle.Location = e.Location;
            Cursor = _cursor2;
        }
        else
        {
            _selecting = false;

            _selectedRectangle.Width = e.X - _selectedRectangle.X;
            _selectedRectangle.Height = e.Y - _selectedRectangle.Y

            Close();
        }
    }

    protected override void OnKeyDown(KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Escape)
        {
            DialogResult = DialogResult.Abort;
        }

        Close();
    }
}
}

```

SettingsForm.Designer.cs

```

namespace ScreenCapturing
{
    partial class SettingsForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}

```

```
}
```

```
#region Windows Form Designer generated code
```

```
/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.components = new System.ComponentModel.Container();
    System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(this);
    this.btnCancel = new System.Windows.Forms.Button();
    this.btnOk = new System.Windows.Forms.Button();
    this.groupBox1 = new System.Windows.Forms.GroupBox();
    this.cbResizeVideo = new System.Windows.Forms.CheckBox();
    this.cbKeepAspectRatio = new System.Windows.Forms.CheckBox();
    this.label6 = new System.Windows.Forms.Label();
    this.label5 = new System.Windows.Forms.Label();
    this.tbHeight = new System.Windows.Forms.TextBox();
    this.tbWidth = new System.Windows.Forms.TextBox();
    this.groupBox4 = new System.Windows.Forms.GroupBox();
    this.linkLabelLog = new System.Windows.Forms.LinkLabel();
    this.cbLog = new System.Windows.Forms.CheckBox();
    this.cmbFPS = new System.Windows.Forms.ComboBox();
    this.label4 = new System.Windows.Forms.Label();
    this.tabControl1 = new System.Windows.Forms.TabControl();
    this.tabPageGeneral = new System.Windows.Forms.TabPage();
    this.groupBox5 = new System.Windows.Forms.GroupBox();
    this.label8 = new System.Windows.Forms.Label();
    this.label14 = new System.Windows.Forms.Label();
    this.cmbWebCameras = new System.Windows.Forms.ComboBox();
    this.tbWebCameraHeight = new System.Windows.Forms.TextBox();
    this.label13 = new System.Windows.Forms.Label();
    this.label15 = new System.Windows.Forms.Label();
    this.tbWebCameraWidth = new System.Windows.Forms.TextBox();
    this.tbWebCameraY = new System.Windows.Forms.TextBox();
    this.label16 = new System.Windows.Forms.Label();
    this.tbWebCameraX = new System.Windows.Forms.TextBox();
    this.label17 = new System.Windows.Forms.Label();
    this.groupBox2 = new System.Windows.Forms.GroupBox();
    this.tabControl2 = new System.Windows.Forms.TabControl();
    this.tabPageWmv = new System.Windows.Forms.TabPage();
    this.cmbWmvAudioFormats = new System.Windows.Forms.ComboBox();
    this.label12 = new System.Windows.Forms.Label();
    this.cmbWmvAudioCodecs = new System.Windows.Forms.ComboBox();
    this.label10 = new System.Windows.Forms.Label();
    this.cmbWmvVideoCodecs = new System.Windows.Forms.ComboBox();
    this.label11 = new System.Windows.Forms.Label();
    this.tabPageAvi = new System.Windows.Forms.TabPage();
    this.cmbAviAudioCodecs = new System.Windows.Forms.ComboBox();
    this.btnVideoCodecProps = new System.Windows.Forms.Button();
    this.label3 = new System.Windows.Forms.Label();
    this.cmbAviVideoCodecs = new System.Windows.Forms.ComboBox();
    this.label2 = new System.Windows.Forms.Label();
    this.groupBox3 = new System.Windows.Forms.GroupBox();
    this.cmbAudioDevices = new System.Windows.Forms.ComboBox();
    this.label9 = new System.Windows.Forms.Label();
    this.cmbAudioLines = new System.Windows.Forms.ComboBox();
    this.label1 = new System.Windows.Forms.Label();
```

```
this.cbEnableAudio = new System.Windows.Forms.CheckBox();
this.tabPageExtra = new System.Windows.Forms.TabPage();
this.grpMouse = new System.Windows.Forms.GroupBox();
this.label24 = new System.Windows.Forms.Label();
this.ccMouseCursorRightClickAnimationColor = new ScreenCapturing.ColorControl();
this.label23 = new System.Windows.Forms.Label();
this.ccMouseCursorLeftClickAnimationColor = new ScreenCapturing.ColorControl();
this.label22 = new System.Windows.Forms.Label();
this.ccMouseHotSpotColor = new ScreenCapturing.ColorControl();
this.label20 = new System.Windows.Forms.Label();
this.label21 = new System.Windows.Forms.Label();
this.tbHotSpotRadius = new System.Windows.Forms.TextBox();
this.label19 = new System.Windows.Forms.Label();
this.label18 = new System.Windows.Forms.Label();
this.tbMouseAnimationDuration = new System.Windows.Forms.TextBox();
this.cbAnimateMouseButtons = new System.Windows.Forms.CheckBox();
this.cbAnimateMouseClicks = new System.Windows.Forms.CheckBox();
this.cbCaptureMouseCursor = new System.Windows.Forms.CheckBox();
this.cbShowMouseHotSpot = new System.Windows.Forms.CheckBox();
this.tabPageAbout = new System.Windows.Forms.TabPage();
this.linkLabel1 = new System.Windows.Forms.LinkLabel();
this.label7 = new System.Windows.Forms.Label();
this.lblProductVersion = new System.Windows.Forms.Label();
this.lblProductName = new System.Windows.Forms.Label();
this.pictureBox1 = new System.Windows.Forms.PictureBox();
this.toolTip1 = new System.Windows.Forms.ToolTip(this.components);
this.groupBox6 = new System.Windows.Forms.GroupBox();
this.label25 = new System.Windows.Forms.Label();
this.ccFrameColor = new ScreenCapturing.ColorControl();
this.label26 = new System.Windows.Forms.Label();
this.label27 = new System.Windows.Forms.Label();
this.tbFrameWidth = new System.Windows.Forms.TextBox();
this.cmbFrameType = new System.Windows.Forms.ComboBox();
this.label28 = new System.Windows.Forms.Label();
this.groupBox1.SuspendLayout();
this.groupBox4.SuspendLayout();
this.tabControl1.SuspendLayout();
this.tabPageGeneral.SuspendLayout();
this.groupBox5.SuspendLayout();
this.groupBox2.SuspendLayout();
this.tabControl2.SuspendLayout();
this.tabPageWmv.SuspendLayout();
this.tabPageAvi.SuspendLayout();
this.groupBox3.SuspendLayout();
this.tabPageExtra.SuspendLayout();
this.grpMouse.SuspendLayout();
this.tabPageAbout.SuspendLayout();
((System.ComponentModel.ISupportInitialize) (this.pictureBox1)).BeginInit();
this.groupBox6.SuspendLayout();
this.SuspendLayout();
//
// btnCancel
//
this.btnCancel.Anchor = ((System.Windows.Forms.AnchorStyles) (System.Windows.Forms.AnchorStyles.Bottom | System.Windows.Forms.AnchorStyles.Right));
this.btnCancel.DialogResult = System.Windows.Forms.DialogResult.Cancel;
this.btnCancel.Location = new System.Drawing.Point(370, 528);
this.btnCancel.Name = "btnCancel";
this.btnCancel.Size = new System.Drawing.Size(75, 23);
this.btnCancel.TabIndex = 2;
this.btnCancel.Text = "Cancel";
```

```
this.btnCancel.UseVisualStyleBackColor = true;
//
// btnOk
//
this.btnOk.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.btnOk.DialogResult = System.Windows.Forms.DialogResult.OK;
this.btnOk.Location = new System.Drawing.Point(289, 528);
this.btnOk.Name = "btnOk";
this.btnOk.Size = new System.Drawing.Size(75, 23);
this.btnOk.TabIndex = 1;
this.btnOk.Text = "OK";
this.btnOk.UseVisualStyleBackColor = true;
this.btnOk.Click += new System.EventHandler(this.btnOk_Click);
//
// groupBox1
//
this.groupBox1.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.groupBox1.Controls.Add(this.cbResizeVideo);
this.groupBox1.Controls.Add(this.cbKeepAspectRatio);
this.groupBox1.Controls.Add(this.label6);
this.groupBox1.Controls.Add(this.label5);
this.groupBox1.Controls.Add(this.tbHeight);
this.groupBox1.Controls.Add(this.tbWidth);
this.groupBox1.Location = new System.Drawing.Point(3, 345);
this.groupBox1.Name = "groupBox1";
this.groupBox1.Size = new System.Drawing.Size(413, 74);
this.groupBox1.TabIndex = 3;
this.groupBox1.TabStop = false;
this.groupBox1.Text = "Video Size";
//
// cbResizeVideo
//
this.cbResizeVideo.AutoSize = true;
this.cbResizeVideo.Location = new System.Drawing.Point(9, 19);
this.cbResizeVideo.Name = "cbResizeVideo";
this.cbResizeVideo.Size = new System.Drawing.Size(382, 17);
this.cbResizeVideo.TabIndex = 0;
this.cbResizeVideo.Text = "Resize to following video size (other";
this.cbResizeVideo.UseVisualStyleBackColor = true;
this.cbResizeVideo.CheckedChanged += new System.EventHandler(this.cbResizeVideo_CheckedChanged);
//
// cbKeepAspectRatio
//
this.cbKeepAspectRatio.AutoSize = true;
this.cbKeepAspectRatio.Location = new System.Drawing.Point(218, 19);
this.cbKeepAspectRatio.Name = "cbKeepAspectRatio";
this.cbKeepAspectRatio.Size = new System.Drawing.Size(109, 17);
this.cbKeepAspectRatio.TabIndex = 3;
this.cbKeepAspectRatio.Text = "Keep aspect ratio";
this.cbKeepAspectRatio.UseVisualStyleBackColor = true;
//
// label6
//
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(102, 45);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(38, 13);
this.label6.TabIndex = 9;
this.label6.Text = "Height";
```

```
//  
// label5  
//  
this.label5.AutoSize = true;  
this.label5.Location = new System.Drawing.Point(5, 45);  
this.label5.Name = "label5";  
this.label5.Size = new System.Drawing.Size(35, 13);  
this.label5.TabIndex = 8;  
this.label5.Text = "Width";  
//  
// tbHeight  
//  
this.tbHeight.Location = new System.Drawing.Point(146, 42);  
this.tbHeight.Name = "tbHeight";  
this.tbHeight.Size = new System.Drawing.Size(50, 20);  
this.tbHeight.TabIndex = 2;  
//  
// tbWidth  
//  
this.tbWidth.Location = new System.Drawing.Point(46, 42);  
this.tbWidth.Name = "tbWidth";  
this.tbWidth.Size = new System.Drawing.Size(50, 20);  
this.tbWidth.TabIndex = 1;  
//  
// groupBox4  
//  
this.groupBox4.Anchor = ((System.Windows.Forms.AnchorStyles)((  
    | System.Windows.Forms.AnchorStyles.Left  
    | System.Windows.Forms.AnchorStyles.Right));  
this.groupBox4.Controls.Add(this.linkLabelLog);  
this.groupBox4.Controls.Add(this.comboBoxLog);  
this.groupBox4.Controls.Add(this.comboBoxFPS);  
this.groupBox4.Controls.Add(this.label4);  
this.groupBox4.Location = new System.Drawing.Point(6, 425);  
this.groupBox4.Name = "groupBox4";  
this.groupBox4.Size = new System.Drawing.Size(413, 53);  
this.groupBox4.TabIndex = 4;  
this.groupBox4.TabStop = false;  
this.groupBox4.Text = "Misc";  
//  
// linkViewLog  
//  
this.linkLabelLog.Anchor = ((System.Windows.Forms.AnchorStyles)((  
    | System.Windows.Forms.AnchorStyles.Left  
    | System.Windows.Forms.AnchorStyles.Right));  
this.linkLabelLog.AutoSize = true;  
this.linkLabelLog.Location = new System.Drawing.Point(356, 23);  
this.linkLabelLog.Name = "linkViewLog";  
this.linkLabelLog.Size = new System.Drawing.Size(47, 13);  
this.linkLabelLog.TabIndex = 4;  
this.linkLabelLog.TabStop = true;  
this.linkLabelLog.Text = "View log";  
this.linkLabelLog.LinkClicked += new System.Windows.Forms.LinkClickedEventHandler(this.linkLabelLog_LinkClicked);  
//  
// comboBoxLog  
//  
this.comboBoxLog.Anchor = ((System.Windows.Forms.AnchorStyles)((  
    | System.Windows.Forms.AnchorStyles.Left  
    | System.Windows.Forms.AnchorStyles.Right));  
this.comboBoxLog.AutoSize = true;  
this.comboBoxLog.Location = new System.Drawing.Point(274, 22);  
this.comboBoxLog.Name = "comboBoxLog";  
this.comboBoxLog.Size = new System.Drawing.Size(68, 17);  
this.comboBoxLog.TabIndex = 1;
```

```
this.cbLog.Text = "Write log";
this.cbLog.UseVisualStyleBackColor = true;
//
// cmbFPS
//
this.cmbFPS.FormattingEnabled = true;
this.cmbFPS.Location = new System.Drawing.Point(50, 19);
this.cmbFPS.Name = "cmbFPS";
this.cmbFPS.Size = new System.Drawing.Size(73, 21);
this.cmbFPS.TabIndex = 0;
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(6, 22);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(27, 13);
this.label4.TabIndex = 3;
this.label4.Text = "FPS";
//
// tabControl1
//
this.tabControl1.Anchor = ((System.Windows.Forms.AnchorStyles)
    | System.Windows.Forms.AnchorStyles.Left
    | System.Windows.Forms.AnchorStyles.Right);
this.tabControl1.Controls.Add(this.tabPageGeneral);
this.tabControl1.Controls.Add(this.tabPageExtra);
this.tabControl1.Controls.Add(this.tabPageAbout);
this.tabControl1.Location = new System.Drawing.Point(12, 12);
this.tabControl1.Name = "tabControl1";
this.tabControl1.SelectedIndex = 0;
this.tabControl1.Size = new System.Drawing.Size(433, 510);
this.tabControl1.TabIndex = 0;
//
// tabPageGeneral
//
this.tabPageGeneral.Controls.Add(this.groupBox5);
this.tabPageGeneral.Controls.Add(this.groupBox2);
this.tabPageGeneral.Controls.Add(this.groupBox3);
this.tabPageGeneral.Controls.Add(this.groupBox4);
this.tabPageGeneral.Controls.Add(this.groupBox1);
this.tabPageGeneral.Location = new System.Drawing.Point(4, 22);
this.tabPageGeneral.Name = "tabPageGeneral";
this.tabPageGeneral.Padding = new System.Windows.Forms.Padding(3);
this.tabPageGeneral.Size = new System.Drawing.Size(425, 484);
this.tabPageGeneral.TabIndex = 0;
this.tabPageGeneral.Text = "General";
this.tabPageGeneral.UseVisualStyleBackColor = true;
//
// groupBox5
//
this.groupBox5.Anchor = ((System.Windows.Forms.AnchorStyles)
    | System.Windows.Forms.AnchorStyles.Right);
this.groupBox5.Controls.Add(this.label8);
this.groupBox5.Controls.Add(this.label14);
this.groupBox5.Controls.Add(this.cmbWebCameras);
this.groupBox5.Controls.Add(this.tbWebCameraHeight);
this.groupBox5.Controls.Add(this.label13);
this.groupBox5.Controls.Add(this.label15);
this.groupBox5.Controls.Add(this.tbWebCameraWidth);
```

```
this.groupBox5.Controls.Add(this.tbWebCameraY);
this.groupBox5.Controls.Add(this.label16);
this.groupBox5.Controls.Add(this.tbWebCameraX);
this.groupBox5.Controls.Add(this.label17);
this.groupBox5.Location = new System.Drawing.Point(6, 113);
this.groupBox5.Name = "groupBox5";
this.groupBox5.Size = new System.Drawing.Size(413, 77);
this.groupBox5.TabIndex = 1;
this.groupBox5.TabStop = false;
this.groupBox5.Text = "Web Camera";
//
// label8
//
this.label8.AutoSize = true;
this.label8.Location = new System.Drawing.Point(6, 48);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(46, 13);
this.label8.TabIndex = 19;
this.label8.Text = "Window";
//
// label14
//
this.label14.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.label14.AutoSize = true;
this.label14.Location = new System.Drawing.Point(261, 48);
this.label14.Name = "label14";
this.label14.Size = new System.Drawing.Size(38, 13);
this.label14.TabIndex = 18;
this.label14.Text = "Height";
this.label14.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// cmbWebCameras
//
this.cmbWebCameras.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.cmbWebCameras.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmbWebCameras.FormattingEnabled = true;
this.cmbWebCameras.Location = new System.Drawing.Point(69, 19);
this.cmbWebCameras.Name = "cmbWebCameras";
this.cmbWebCameras.Size = new System.Drawing.Size(338, 21);
this.cmbWebCameras.TabIndex = 1;
this.toolTip1.SetToolTip(this.cmbWebCameras, resources.GetString("$T{ToolTip1}"));
//
// tbWebCameraHeight
//
this.tbWebCameraHeight.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.tbWebCameraHeight.Location = new System.Drawing.Point(305, 19);
this.tbWebCameraHeight.Name = "tbWebCameraHeight";
this.tbWebCameraHeight.Size = new System.Drawing.Size(32, 20);
this.tbWebCameraHeight.TabIndex = 5;
this.tbWebCameraHeight.Text = "120";
//
// label13
//
this.label13.AutoSize = true;
this.label13.Location = new System.Drawing.Point(6, 22);
this.label13.Name = "label13";
this.label13.Size = new System.Drawing.Size(41, 13);
this.label13.TabIndex = 5;
this.label13.Text = "Device";
```

```
//  
// label15  
//  
this.label15.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right));  
this.label15.AutoSize = true;  
this.label15.Location = new System.Drawing.Point(182, 48);  
this.label15.Name = "label15";  
this.label15.Size = new System.Drawing.Size(35, 13);  
this.label15.TabIndex = 17;  
this.label15.Text = "Width";  
this.label15.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;  
//  
// tbWebCameraWidth  
//  
this.tbWebCameraWidth.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right));  
this.tbWebCameraWidth.Location = new System.Drawing.Point(223, 48);  
this.tbWebCameraWidth.Name = "tbWebCameraWidth";  
this.tbWebCameraWidth.Size = new System.Drawing.Size(32, 20);  
this.tbWebCameraWidth.TabIndex = 4;  
this.tbWebCameraWidth.Text = "160";  
//  
// tbWebCameraY  
//  
this.tbWebCameraY.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right));  
this.tbWebCameraY.Location = new System.Drawing.Point(144, 45);  
this.tbWebCameraY.Name = "tbWebCameraY";  
this.tbWebCameraY.Size = new System.Drawing.Size(32, 20);  
this.tbWebCameraY.TabIndex = 3;  
this.tbWebCameraY.Text = "10";  
//  
// label16  
//  
this.label16.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right));  
this.label16.AutoSize = true;  
this.label16.Location = new System.Drawing.Point(124, 48);  
this.label16.Name = "label16";  
this.label16.Size = new System.Drawing.Size(14, 13);  
this.label16.TabIndex = 16;  
this.label16.Text = "Y";  
this.label16.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;  
//  
// tbWebCameraX  
//  
this.tbWebCameraX.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right));  
this.tbWebCameraX.Location = new System.Drawing.Point(86, 45);  
this.tbWebCameraX.Name = "tbWebCameraX";  
this.tbWebCameraX.Size = new System.Drawing.Size(32, 20);  
this.tbWebCameraX.TabIndex = 2;  
this.tbWebCameraX.Text = "10";  
//  
// label17  
//  
this.label17.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right));  
this.label17.AutoSize = true;  
this.label17.Location = new System.Drawing.Point(66, 48);  
this.label17.Name = "label17";  
this.label17.Size = new System.Drawing.Size(14, 13);  
this.label17.TabIndex = 13;  
this.label17.Text = "X";  
this.label17.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;
```

```
//  
// groupBox2  
//  
this.groupBox2.Anchor = ((System.Windows.Forms.AnchorStyles)((  
    System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right));  
this.groupBox2.Controls.Add(this.tabControl2);  
this.groupBox2.Location = new System.Drawing.Point(3, 196);  
this.groupBox2.Name = "groupBox2";  
this.groupBox2.Size = new System.Drawing.Size(413, 143);  
this.groupBox2.TabIndex = 2;  
this.groupBox2.TabStop = false;  
this.groupBox2.Text = "Output Video Format (Select tab to choose format)  
//  
// tabControl2  
//  
this.tabControl2.Anchor = ((System.Windows.Forms.AnchorStyles)((  
    System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right));  
this.tabControl2.Controls.Add(this.tabPageWmv);  
this.tabControl2.Controls.Add(this.tabPageAvi);  
this.tabControl2.Location = new System.Drawing.Point(6, 19);  
this.tabControl2.Name = "tabControl2";  
this.tabControl2.SelectedIndex = 0;  
this.tabControl2.Size = new System.Drawing.Size(401, 116);  
this.tabControl2.TabIndex = 8;  
//  
// tabPageWmv  
//  
this.tabPageWmv.Controls.Add(this.cmbWmvAudioFormats);  
this.tabPageWmv.Controls.Add(this.label12);  
this.tabPageWmv.Controls.Add(this.cmbWmvAudioCodecs);  
this.tabPageWmv.Controls.Add(this.label10);  
this.tabPageWmv.Controls.Add(this.cmbWmvVideoCodecs);  
this.tabPageWmv.Controls.Add(this.label11);  
this.tabPageWmv.Location = new System.Drawing.Point(4, 22);  
this.tabPageWmv.Name = "tabPageWmv";  
this.tabPageWmv.Padding = new System.Windows.Forms.Padding(3);  
this.tabPageWmv.Size = new System.Drawing.Size(393, 90);  
this.tabPageWmv.TabIndex = 1;  
this.tabPageWmv.Text = "WMV Format";  
this.tabPageWmv.UseVisualStyleBackColor = true;  
//  
// cmbWmvAudioFormats  
//  
this.cmbWmvAudioFormats.Anchor = ((System.Windows.Forms.AnchorStyles)((  
    System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right));  
this.cmbWmvAudioFormats.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;  
this.cmbWmvAudioFormats.FormattingEnabled = true;  
this.cmbWmvAudioFormats.Location = new System.Drawing.Point(62, 11);  
this.cmbWmvAudioFormats.Name = "cmbWmvAudioFormats";  
this.cmbWmvAudioFormats.Size = new System.Drawing.Size(325, 21);  
this.cmbWmvAudioFormats.TabIndex = 1;  
//  
// label12  
//  
this.label12.AutoSize = true;  
this.label12.Location = new System.Drawing.Point(6, 36);  
this.label12.Name = "label12";  
this.label12.Size = new System.Drawing.Size(39, 13);  
this.label12.TabIndex = 14;  
this.label12.Text = "Format";
```

```
//  
// cmbWmvAudioCodecs  
//  
this.cmbWmvAudioCodecs.Anchor = ((System.Windows.Forms.AnchorStyles)  
    | System.Windows.Forms.AnchorStyles.Right);  
this.cmbWmvAudioCodecs.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;  
this.cmbWmvAudioCodecs.FormattingEnabled = true;  
this.cmbWmvAudioCodecs.Location = new System.Drawing.Point(62, 180);  
this.cmbWmvAudioCodecs.Name = "cmbWmvAudioCodecs";  
this.cmbWmvAudioCodecs.Size = new System.Drawing.Size(325, 21);  
this.cmbWmvAudioCodecs.TabIndex = 0;  
this.cmbWmvAudioCodecs.SelectedIndexChanged += new System.EventHandler(this.cmbWmvAudioCodecs_SelectedIndexChanged);  
//  
// label10  
//  
this.label10.AutoSize = true;  
this.label10.Location = new System.Drawing.Point(6, 63);  
this.label10.Name = "label10";  
this.label10.Size = new System.Drawing.Size(34, 13);  
this.label10.TabIndex = 12;  
this.label10.Text = "Video";  
//  
// cmbWmvVideoCodecs  
//  
this.cmbWmvVideoCodecs.Anchor = ((System.Windows.Forms.AnchorStyles)  
    | System.Windows.Forms.AnchorStyles.Right);  
this.cmbWmvVideoCodecs.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;  
this.cmbWmvVideoCodecs.FormattingEnabled = true;  
this.cmbWmvVideoCodecs.Location = new System.Drawing.Point(62, 210);  
this.cmbWmvVideoCodecs.Name = "cmbWmvVideoCodecs";  
this.cmbWmvVideoCodecs.Size = new System.Drawing.Size(325, 21);  
this.cmbWmvVideoCodecs.TabIndex = 2;  
//  
// label11  
//  
this.label11.AutoSize = true;  
this.label11.Location = new System.Drawing.Point(6, 9);  
this.label11.Name = "label11";  
this.label11.Size = new System.Drawing.Size(34, 13);  
this.label11.TabIndex = 11;  
this.label11.Text = "Audio";  
//  
// tabPageAvi  
//  
this.tabPageAvi.Controls.Add(this.cmbAviAudioCodecs);  
this.tabPageAvi.Controls.Add(this.btnVideoCodecProps);  
this.tabPageAvi.Controls.Add(this.label3);  
this.tabPageAvi.Controls.Add(this.cmbAviVideoCodecs);  
this.tabPageAvi.Controls.Add(this.label2);  
this.tabPageAvi.Location = new System.Drawing.Point(4, 22);  
this.tabPageAvi.Name = "tabPageAvi";  
this.tabPageAvi.Padding = new System.Windows.Forms.Padding(3);  
this.tabPageAvi.Size = new System.Drawing.Size(393, 90);  
this.tabPageAvi.TabIndex = 2;  
this.tabPageAvi.Text = "AVI format";  
this.tabPageAvi.UseVisualStyleBackColor = true;  
//  
// cmbAviAudioCodecs  
//  
this.cmbAviAudioCodecs.Anchor = ((System.Windows.Forms.AnchorStyles)  
    | System.Windows.Forms.AnchorStyles.Right);  
this.cmbAviAudioCodecs.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
```

```
    | System.Windows.Forms.AnchorStyles.Right);
this.cmbAviAudioCodecs.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmbAviAudioCodecs.FormattingEnabled = true;
this.cmbAviAudioCodecs.Location = new System.Drawing.Point(49, 100);
this.cmbAviAudioCodecs.Name = "cmbAviAudioCodecs";
this.cmbAviAudioCodecs.Size = new System.Drawing.Size(305, 21);
this.cmbAviAudioCodecs.TabIndex = 0;
//
// btnVideoCodecProps
//
this.btnVideoCodecProps.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.btnVideoCodecProps.Image = global::ScreenCapturing.Properties.Resources.video_codec_props;
this.btnVideoCodecProps.Location = new System.Drawing.Point(360, 100);
this.btnVideoCodecProps.Name = "btnVideoCodecProps";
this.btnVideoCodecProps.Size = new System.Drawing.Size(26, 26);
this.btnVideoCodecProps.TabIndex = 2;
this.btnVideoCodecProps.UseVisualStyleBackColor = true;
this.btnVideoCodecProps.Click += new System.EventHandler(this.btnVideoCodecProps_Click);
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(5, 49);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(34, 13);
this.label3.TabIndex = 7;
this.label3.Text = "Video";
//
// cmbAviVideoCodecs
//
this.cmbAviVideoCodecs.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.cmbAviVideoCodecs.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmbAviVideoCodecs.FormattingEnabled = true;
this.cmbAviVideoCodecs.Location = new System.Drawing.Point(49, 130);
this.cmbAviVideoCodecs.Name = "cmbAviVideoCodecs";
this.cmbAviVideoCodecs.Size = new System.Drawing.Size(305, 21);
this.cmbAviVideoCodecs.TabIndex = 1;
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(5, 22);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(34, 13);
this.label2.TabIndex = 5;
this.label2.Text = "Audio";
//
// groupBox3
//
this.groupBox3.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.groupBox3.Controls.Add(this.cmbAudioDevices);
this.groupBox3.Controls.Add(this.label9);
this.groupBox3.Controls.Add(this.cmbAudioLines);
this.groupBox3.Controls.Add(this.label1);
this.groupBox3.Controls.Add(this.cbEnableAudio);
this.groupBox3.Location = new System.Drawing.Point(6, 6);
this.groupBox3.Name = "groupBox3";
this.groupBox3.Size = new System.Drawing.Size(413, 101);
```

```
this groupBox3.TabIndex = 0;
this.groupBox3.TabStop = false;
this.groupBox3.Text = "Audio Input";
//
// cmbAudioDevices
//
this.cmbAudioDevices.Anchor = ((System.Windows.Forms.AnchorStyles)
    | System.Windows.Forms.AnchorStyles.Right);
this.cmbAudioDevices.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmbAudioDevices.FormattingEnabled = true;
this.cmbAudioDevices.Location = new System.Drawing.Point(69, 42);
this.cmbAudioDevices.Name = "cmbAudioDevices";
this.cmbAudioDevices.Size = new System.Drawing.Size(338, 21);
this.cmbAudioDevices.TabIndex = 1;
this.cmbAudioDevices.SelectedIndexChanged += new System.EventHandler(this.cmbAudioDevices_SelectedIndexChanged);
//
// label9
//
this.label9.AutoSize = true;
this.label9.Location = new System.Drawing.Point(6, 45);
this.label9.Name = "label9";
this.label9.Size = new System.Drawing.Size(41, 13);
this.label9.TabIndex = 5;
this.label9.Text = "Device";
//
// cmbAudioLines
//
this.cmbAudioLines.Anchor = ((System.Windows.Forms.AnchorStyles)
    | System.Windows.Forms.AnchorStyles.Right);
this.cmbAudioLines.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmbAudioLines.FormattingEnabled = true;
this.cmbAudioLines.Location = new System.Drawing.Point(69, 67);
this.cmbAudioLines.Name = "cmbAudioLines";
this.cmbAudioLines.Size = new System.Drawing.Size(338, 21);
this.cmbAudioLines.TabIndex = 2;
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(6, 70);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(27, 13);
this.label1.TabIndex = 3;
this.label1.Text = "Line";
//
// cbEnableAudio
//
this.cbEnableAudio.AutoSize = true;
this.cbEnableAudio.Checked = true;
this.cbEnableAudio.CheckState = System.Windows.Forms.CheckState.Checked;
this.cbEnableAudio.Location = new System.Drawing.Point(9, 19);
this.cbEnableAudio.Name = "cbEnableAudio";
this.cbEnableAudio.Size = new System.Drawing.Size(89, 17);
this.cbEnableAudio.TabIndex = 0;
this.cbEnableAudio.Text = "Enable Audio";
this.cbEnableAudio.UseVisualStyleBackColor = true;
this.cbEnableAudio.CheckedChanged += new System.EventHandler(this.cbEnableAudio_CheckedChanged);
//
// tabPageExtra
//

```

```
this.tabPageExtra.Controls.Add(this.groupBox6);
this.tabPageExtra.Controls.Add(this.grpMouse);
this.tabPageExtra.Location = new System.Drawing.Point(4, 22);
this.tabPageExtra.Name = "tabPageExtra";
this.tabPageExtra.Padding = new System.Windows.Forms.Padding(3);
this.tabPageExtra.Size = new System.Drawing.Size(425, 484);
this.tabPageExtra.TabIndex = 2;
this.tabPageExtra.Text = "Extra";
this.tabPageExtra.UseVisualStyleBackColor = true;
//
// grpMouse
//
this.grpMouse.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.grpMouse.Controls.Add(this.label24);
this.grpMouse.Controls.Add(this.ccMouseCursorRightClickAnimationColor);
this.grpMouse.Controls.Add(this.label23);
this.grpMouse.Controls.Add(this.ccMouseCursorLeftClickAnimationColor);
this.grpMouse.Controls.Add(this.label22);
this.grpMouse.Controls.Add(this.ccMouseHotSpotColor);
this.grpMouse.Controls.Add(this.label20);
this.grpMouse.Controls.Add(this.label21);
this.grpMouse.Controls.Add(this.tbHotSpotRadius);
this.grpMouse.Controls.Add(this.label19);
this.grpMouse.Controls.Add(this.label18);
this.grpMouse.Controls.Add(this.tbMouseAnimationDuration);
this.grpMouse.Controls.Add(this.cbAnimateMouseButtons);
this.grpMouse.Controls.Add(this.cbAnimateMouseClicks);
this.grpMouse.Controls.Add(this.cbCaptureMouseCursor);
this.grpMouse.Controls.Add(this.cbShowMouseHotSpot);
this.grpMouse.Location = new System.Drawing.Point(6, 6);
this.grpMouse.Name = "grpMouse";
this.grpMouse.Size = new System.Drawing.Size(413, 259);
this.grpMouse.TabIndex = 0;
this.grpMouse.TabStop = false;
this.grpMouse.Text = "Mouse cursor";
//
// label24
//
this.label24.AutoSize = true;
this.label24.Location = new System.Drawing.Point(6, 231);
this.label24.Name = "label24";
this.label24.Size = new System.Drawing.Size(131, 13);
this.label24.TabIndex = 15;
this.label24.Text = "Right click animation color";
//
// ccMouseCursorRightClickAnimationColor
//
this.ccMouseCursorRightClickAnimationColor.BorderStyle = ScreenBrushStyle.FixedSingle;
this.ccMouseCursorRightClickAnimationColor.Location = new System.Drawing.Point(6, 251);
this.ccMouseCursorRightClickAnimationColor.Name = "ccMouseCursorRightClickAnimationColor";
this.ccMouseCursorRightClickAnimationColor.Size = new System.Drawing.Size(124, 13);
this.ccMouseCursorRightClickAnimationColor.TabIndex = 8;
//
// label23
//
this.label23.AutoSize = true;
this.label23.Location = new System.Drawing.Point(6, 205);
this.label23.Name = "label23";
this.label23.Size = new System.Drawing.Size(124, 13);
```

```
this.label23.TabIndex = 13;
this.label23.Text = "Left click animation color";
//
// ccMouseCursorLeftClickAnimationColor
//
this.ccMouseCursorLeftClickAnimationColor.BorderStyle = ScreenCapturing.CustomBorderStyle;
this.ccMouseCursorLeftClickAnimationColor.Location = new System.Drawing.Point(148, 125);
this.ccMouseCursorLeftClickAnimationColor.Name = "ccMouseCursorLeftClickAnimationColor";
this.ccMouseCursorLeftClickAnimationColor.Size = new System.Drawing.Size(140, 20);
this.ccMouseCursorLeftClickAnimationColor.TabIndex = 7;
//
// label22
//
this.label22.AutoSize = true;
this.label22.Location = new System.Drawing.Point(6, 125);
this.label22.Name = "label22";
this.label22.Size = new System.Drawing.Size(73, 13);
this.label22.TabIndex = 11;
this.label22.Text = "Hot spot color";
//
// ccMouseHotSpotColor
//
this.ccMouseHotSpotColor.BorderStyle = ScreenCapturing.CustomBorderStyle;
this.ccMouseHotSpotColor.Location = new System.Drawing.Point(148, 148);
this.ccMouseHotSpotColor.Name = "ccMouseHotSpotColor";
this.ccMouseHotSpotColor.Size = new System.Drawing.Size(38, 20);
this.ccMouseHotSpotColor.TabIndex = 4;
//
// label20
//
this.label20.AutoSize = true;
this.label20.Location = new System.Drawing.Point(192, 99);
this.label20.Name = "label20";
this.label20.Size = new System.Drawing.Size(33, 13);
this.label20.TabIndex = 9;
this.label20.Text = "pixels";
//
// label21
//
this.label21.AutoSize = true;
this.label21.Location = new System.Drawing.Point(6, 99);
this.label21.Name = "label21";
this.label21.Size = new System.Drawing.Size(78, 13);
this.label21.TabIndex = 8;
this.label21.Text = "Hot spot radius";
//
// tbHotSpotRadius
//
this.tbHotSpotRadius.Location = new System.Drawing.Point(148, 99);
this.tbHotSpotRadius.Name = "tbHotSpotRadius";
this.tbHotSpotRadius.Size = new System.Drawing.Size(38, 20);
this.tbHotSpotRadius.TabIndex = 3;
//
// label19
//
this.label19.AutoSize = true;
this.label19.Location = new System.Drawing.Point(192, 73);
this.label19.Name = "label19";
this.label19.Size = new System.Drawing.Size(20, 13);
this.label19.TabIndex = 6;
```

```
this.label19.Text = "ms";
//
// label18
//
this.label18.AutoSize = true;
this.label18.Location = new System.Drawing.Point(6, 73);
this.label18.Name = "label18";
this.label18.Size = new System.Drawing.Size(136, 13);
this.label18.TabIndex = 5;
this.label18.Text = "Hot spot animation duration";
//
// tbMouseAnimationDuration
//
this.tbMouseAnimationDuration.Location = new System.Drawing.Point(6, 93);
this.tbMouseAnimationDuration.Name = "tbMouseAnimationDuration";
this.tbMouseAnimationDuration.Size = new System.Drawing.Size(38, 20);
this.tbMouseAnimationDuration.TabIndex = 2;
//
// cbAnimateMouseButtons
//
this.cbAnimateMouseButtons.AutoSize = true;
this.cbAnimateMouseButtons.Location = new System.Drawing.Point(6, 113);
this.cbAnimateMouseButtons.Name = "cbAnimateMouseButtons";
this.cbAnimateMouseButtons.Size = new System.Drawing.Size(136, 13);
this.cbAnimateMouseButtons.TabIndex = 6;
this.cbAnimateMouseButtons.Text = "Animate mouse buttons";
this.cbAnimateMouseButtons.UseVisualStyleBackColor = true;
//
// cbAnimateMouseClicks
//
this.cbAnimateMouseClicks.AutoSize = true;
this.cbAnimateMouseClicks.Location = new System.Drawing.Point(6, 133);
this.cbAnimateMouseClicks.Name = "cbAnimateMouseClicks";
this.cbAnimateMouseClicks.Size = new System.Drawing.Size(128, 13);
this.cbAnimateMouseClicks.TabIndex = 5;
this.cbAnimateMouseClicks.Text = "Animate mouse clicks";
this.cbAnimateMouseClicks.UseVisualStyleBackColor = true;
//
// cbCaptureMouseCursor
//
this.cbCaptureMouseCursor.AutoSize = true;
this.cbCaptureMouseCursor.Location = new System.Drawing.Point(6, 153);
this.cbCaptureMouseCursor.Name = "cbCaptureMouseCursor";
this.cbCaptureMouseCursor.Size = new System.Drawing.Size(129, 13);
this.cbCaptureMouseCursor.TabIndex = 0;
this.cbCaptureMouseCursor.Text = "Capture mouse cursor";
this.cbCaptureMouseCursor.UseVisualStyleBackColor = true;
//
// cbShowMouseHotSpot
//
this.cbShowMouseHotSpot.AutoSize = true;
this.cbShowMouseHotSpot.Location = new System.Drawing.Point(6, 173);
this.cbShowMouseHotSpot.Name = "cbShowMouseHotSpot";
this.cbShowMouseHotSpot.Size = new System.Drawing.Size(128, 17);
this.cbShowMouseHotSpot.TabIndex = 1;
this.cbShowMouseHotSpot.Text = "Show mouse hot spot";
this.cbShowMouseHotSpot.UseVisualStyleBackColor = true;
//
// tabPageAbout
//

```

```
this.tabPageAbout.Controls.Add(this.linkLabel1);
this.tabPageAbout.Controls.Add(this.label7);
this.tabPageAbout.Controls.Add(this.lblProductVersion);
this.tabPageAbout.Controls.Add(this.lblProductName);
this.tabPageAbout.Controls.Add(this.pictureBox1);
this.tabPageAbout.Location = new System.Drawing.Point(4, 22);
this.tabPageAbout.Name = "tabPageAbout";
this.tabPageAbout.Padding = new System.Windows.Forms.Padding(3);
this.tabPageAbout.Size = new System.Drawing.Size(425, 484);
this.tabPageAbout.TabIndex = 1;
this.tabPageAbout.Text = "About...";
this.tabPageAbout.UseVisualStyleBackColor = true;
//
// linkLabel1
//
this.linkLabel1.AutoSize = true;
this.linkLabel1.Location = new System.Drawing.Point(53, 72);
this.linkLabel1.Name = "linkLabel1";
this.linkLabel1.Size = new System.Drawing.Size(103, 13);
this.linkLabel1.TabIndex = 8;
this.linkLabel1.TabStop = true;
this.linkLabel1.Text = "www.bytescout.com";
this.linkLabel1.LinkClicked += new System.Windows.Forms.LinkLabelLinkClickedEventHandler(linkLabel1_LinkClicked);
//
// label7
//
this.label7.AutoSize = true;
this.label7.Location = new System.Drawing.Point(53, 50);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(169, 13);
this.label7.TabIndex = 7;
this.label7.Text = "Copyright ♦ 2009-2013 ByteScout";
//
// lblProductVersion
//
this.lblProductVersion.AutoSize = true;
this.lblProductVersion.Location = new System.Drawing.Point(53, 100);
this.lblProductVersion.Name = "lblProductVersion";
this.lblProductVersion.Size = new System.Drawing.Size(164, 13);
this.lblProductVersion.TabIndex = 6;
this.lblProductVersion.Text = "Version _PRODUCT_VERSION_";
//
// lblProductName
//
this.lblProductName.AutoSize = true;
this.lblProductName.Font = new System.Drawing.Font("Tahoma", 8);
this.lblProductName.Location = new System.Drawing.Point(53, 6);
this.lblProductName.Name = "lblProductName";
this.lblProductName.Size = new System.Drawing.Size(112, 13);
this.lblProductName.TabIndex = 5;
this.lblProductName.Text = "_PRODUCT_NAME_";
//
// pictureBox1
//
this.pictureBox1.Image = global::ScreenCapturing.Properties.Resources.icon;
this.pictureBox1.Location = new System.Drawing.Point(6, 6);
this.pictureBox1.Name = "pictureBox1";
this.pictureBox1.Size = new System.Drawing.Size(32, 32);
this.pictureBox1.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;
this.pictureBox1.TabIndex = 0;
```

```
this.pictureBox1.TabStop = false;
//
// groupBox6
//
this.groupBox6.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.groupBox6.Controls.Add(this.label28);
this.groupBox6.Controls.Add(this.cmbFrameType);
this.groupBox6.Controls.Add(this.label26);
this.groupBox6.Controls.Add(this.label25);
this.groupBox6.Controls.Add(this.label27);
this.groupBox6.Controls.Add(this.tbFrameWidth);
this.groupBox6.Controls.Add(this.ccFrameColor);
this.groupBox6.Location = new System.Drawing.Point(6, 271);
this.groupBox6.Name = "groupBox6";
this.groupBox6.Size = new System.Drawing.Size(413, 104);
this.groupBox6.TabIndex = 1;
this.groupBox6.TabStop = false;
this.groupBox6.Text = "Capturing Region Frame";
//
// label25
//
this.label25.AutoSize = true;
this.label25.Location = new System.Drawing.Point(6, 49);
this.label25.Name = "label25";
this.label25.Size = new System.Drawing.Size(62, 13);
this.label25.TabIndex = 17;
this.label25.Text = "Frame color";
//
// ccFrameColor
//
this.ccFrameColor.BorderStyle = ScreenCapturing.CustomBorderStyle.FixedSingle;
this.ccFrameColor.Location = new System.Drawing.Point(148, 46);
this.ccFrameColor.Name = "ccFrameColor";
this.ccFrameColor.Size = new System.Drawing.Size(38, 20);
this.ccFrameColor.TabIndex = 1;
//
// label26
//
this.label26.AutoSize = true;
this.label26.Location = new System.Drawing.Point(192, 75);
this.label26.Name = "label26";
this.label26.Size = new System.Drawing.Size(33, 13);
this.label26.TabIndex = 18;
this.label26.Text = "pixels";
//
// label27
//
this.label27.AutoSize = true;
this.label27.Location = new System.Drawing.Point(6, 75);
this.label27.Name = "label27";
this.label27.Size = new System.Drawing.Size(64, 13);
this.label27.TabIndex = 17;
this.label27.Text = "Frame width";
//
// tbFrameWidth
//
this.tbFrameWidth.Location = new System.Drawing.Point(148, 72);
this.tbFrameWidth.Name = "tbFrameWidth";
this.tbFrameWidth.Size = new System.Drawing.Size(38, 20);
```

```
this.tbFrameWidth.TabIndex = 2;
//
// cmbFrameType
//
this.cmbFrameType.DropDownStyle = System.Windows.Forms.ComboBoxStyle;
this.cmbFrameType.FormattingEnabled = true;
this.cmbFrameType.Location = new System.Drawing.Point(148, 19);
this.cmbFrameType.Name = "cmbFrameType";
this.cmbFrameType.Size = new System.Drawing.Size(166, 21);
this.cmbFrameType.TabIndex = 0;
//
// label28
//
this.label28.AutoSize = true;
this.label28.Location = new System.Drawing.Point(6, 22);
this.label28.Name = "label28";
this.label28.Size = new System.Drawing.Size(59, 13);
this.label28.TabIndex = 19;
this.label28.Text = "Frame type";
//
// SettingsForm
//
this.AcceptButton = this.btnOk;
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.CancelButton = this.btnCancel;
this.ClientSize = new System.Drawing.Size(457, 563);
this.Controls.Add(this.tabControl1);
this.Controls.Add(this.btnOk);
this.Controls.Add(this.btnCancel);
this.MaximizeBox = false;
this.MinimizeBox = false;
this.Name = "SettingsForm";
this.ShowIcon = false;
this.ShowInTaskbar = false;
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;
this.Text = "Settings";
this.groupBox1.ResumeLayout(false);
this.groupBox1.PerformLayout();
this.groupBox4.ResumeLayout(false);
this.groupBox4.PerformLayout();
this.tabControl1.ResumeLayout(false);
this.tabPageGeneral.ResumeLayout(false);
this.tabPageGeneral.PerformLayout();
this.groupBox5.ResumeLayout(false);
this.groupBox5.PerformLayout();
this.groupBox2.ResumeLayout(false);
this.groupBox2.PerformLayout();
this.tabControl2.ResumeLayout(false);
this.tabPageWmv.ResumeLayout(false);
this.tabPageWmv.PerformLayout();
this.tabPageAvi.ResumeLayout(false);
this.tabPageAvi.PerformLayout();
this.groupBox3.ResumeLayout(false);
this.groupBox3.PerformLayout();
this.tabPageExtra.ResumeLayout(false);
this.tabPageExtra.PerformLayout();
this.grpMouse.ResumeLayout(false);
this.grpMouse.PerformLayout();
this.tabPageAbout.ResumeLayout(false);
this.tabPageAbout.PerformLayout();
this.tabPageAbout.PerformLayout();
((System.ComponentModel.ISupportInitialize) (this.pictureBox1))
this.groupBox6.ResumeLayout(false);
```

```
        this.groupBox6.PerformLayout();
        this.ResumeLayout(false);

    }

#endregion

private System.Windows.Forms.Button btnCancel;
private System.Windows.Forms.Button btnOk;
private System.Windows.Forms.GroupBox groupBox1;
private System.Windows.Forms.TextBox tbHeight;
private System.Windows.Forms.TextBox tbWidth;
private System.Windows.Forms.GroupBox groupBox4;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.Label label6;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.CheckBox cbKeepAspectRatio;
private System.Windows.Forms.CheckBox cbResizeVideo;
private System.Windows.Forms.ComboBox cmbFPS;
private System.Windows.Forms.CheckBox cbLog;
private System.Windows.Forms.TabControl tabControl1;
private System.Windows.Forms.TabPage tabPageGeneral;
private System.Windows.Forms.TabPage tabPageAbout;
private System.Windows.Forms.PictureBox pictureBox1;
private System.Windows.Forms.LinkLabel linkLabel1;
private System.Windows.Forms.Label label7;
private System.Windows.Forms.Label lblProductVersion;
private System.Windows.Forms.Label lblProductName;
private System.Windows.Forms.LinkLabel linkViewLog;
private System.Windows.Forms.GroupBox groupBox2;
private System.Windows.Forms.TabControl tabControl2;
private System.Windows.Forms.TabPage tabPageWmv;
private System.Windows.Forms.ComboBox cmbWmvAudioFormats;
private System.Windows.Forms.Label label12;
private System.Windows.Forms.ComboBox cmbWmvAudioCodecs;
private System.Windows.Forms.Label label10;
private System.Windows.Forms.ComboBox cmbWmvVideoCodecs;
private System.Windows.Forms.Label label11;
private System.Windows.Forms.TabPage tabPageAvi;
private System.Windows.Forms.ComboBox cmbAviAudioCodecs;
private System.Windows.Forms.Button btnVideoCodecProps;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.ComboBox cmbAviVideoCodecs;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.GroupBox groupBox3;
private System.Windows.Forms.ComboBox cmbAudioDevices;
private System.Windows.Forms.Label label9;
private System.Windows.Forms.ComboBox cmbAudioLines;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.CheckBox cbEnableAudio;
private System.Windows.Forms.GroupBox groupBox5;
private System.Windows.Forms.Label label14;
private System.Windows.Forms.ComboBox cmbWebCameras;
private System.Windows.Forms.TextBox tbWebCameraHeight;
private System.Windows.Forms.Label label13;
private System.Windows.Forms.Label label15;
private System.Windows.Forms.TextBox tbWebCameraWidth;
private System.Windows.Forms.TextBox tbWebCameraY;
private System.Windows.Forms.Label label16;
private System.Windows.Forms.TextBox tbWebCameraX;
```

```
    private System.Windows.Forms.Label label17;
    private System.Windows.Forms.ToolTip toolTip1;
    private System.Windows.Forms.Label label18;
    private System.Windows.Forms.TabPage tabPageExtra;
    private System.Windows.Forms.GroupBox grpMouse;
    private System.Windows.Forms.CheckBox cbShowMouseHotSpot;
    private System.Windows.Forms.CheckBox cbCaptureMouseCursor;
    private System.Windows.Forms.CheckBox cbAnimateMouseClicks;
    private System.Windows.Forms.CheckBox cbAnimateMouseButtons;
    private System.Windows.Forms.Label label18;
    private System.Windows.Forms.TextBox tbMouseAnimationDuration;
    private System.Windows.Forms.Label label19;
    private System.Windows.Forms.Label label20;
    private System.Windows.Forms.Label label21;
    private System.Windows.Forms.TextBox tbHotSpotRadius;
    private System.Windows.Forms.Label label22;
    private ColorControl ccMouseHotSpotColor;
    private System.Windows.Forms.Label label23;
    private ColorControl ccMouseCursorLeftClickAnimationColor;
    private System.Windows.Forms.Label label24;
    private ColorControl ccMouseCursorRightClickAnimationColor;
    private System.Windows.Forms.GroupBox groupBox6;
    private System.Windows.Forms.Label label25;
    private ColorControl ccFrameColor;
    private System.Windows.Forms.Label label26;
    private System.Windows.Forms.Label label27;
    private System.Windows.Forms.TextBox tbFrameWidth;
    private System.Windows.Forms.ComboBox cmbFrameType;
    private System.Windows.Forms.Label label28;
}
}
```

SettingsForm.cs

```
using System;
using System.Diagnostics;
using System.Globalization;
using System.IO;
using System.Runtime.InteropServices;
using System.Windows.Forms;
using BytescoutScreenCapturingLib;

// NOTE: if you are getting error like "invalid image" related to loading the SDK's DLL
// try to do the following:
// 1) remove the reference to the SDK by View - Solution Explorer
// then click on References, select Bytescout... reference name and right-click it and
// 2) To re-add click on the menu: Project - Add Reference
// 3) In "Add Reference" dialog switch to "COM" tab and find Bytescout...
// 4) Select it and click "Add"
// 5) Recompile the application
```

```
// Note: if you need to run on both x64 and x86 then please make sure you have set "Embeded Resource" to "true" for the ScreenCapturing.dll file

namespace ScreenCapturing
{
    public partial class SettingsForm : Form
    {
        private readonly Capturer _tempCapturer;

        public SettingsForm()
        {
            InitializeComponent();

            lblProductName.Text = Application.ProductName;
            lblProductVersion.Text = "Version " + Application.ProductVersion;

            _tempCapturer = new Capturer();
            _tempCapturer.RegistrationName = "demo";
            _tempCapturer.RegistrationKey = "demo";

            cmbFPS.Items.AddRange(new object[] { 5f, 7.5f, 10f, 12f, 14.985f });

            if (_tempCapturer.AudioDeviceCount == 0)
            {
                cbEnableAudio.Checked = false;
                cbEnableAudio.Enabled = false;

                cmbAudioDevices.Items.Add("No audio devices found.");
                cmbAudioDevices.SelectedIndex = 0;
                cmbAudioDevices.Enabled = false;

                cmbAudioLines.Enabled = false;
            }
            else
            {
                cbEnableAudio.Checked = Program.Cfg.EnableAudio;

                for (int i = 0; i < _tempCapturer.AudioDeviceCount; i++)
                {
                    string device = _tempCapturer.GetAudioDeviceName(i);
                    cmbAudioDevices.Items.Add(device);
                }
            }

            for (int i = 0; i < cmbAudioDevices.Items.Count; i++)
            {
                if (cmbAudioDevices.Items[i].ToString() == Program.Cfg.AudioDevice)
                {
                    cmbAudioDevices.SelectedIndex = i;
                    break;
                }
            }

            if (cmbAudioDevices.SelectedIndex == -1)
            {
                cmbAudioDevices.SelectedItem = _tempCapturer.CurrentAudioDevice;
                Program.Cfg.AudioDevice = _tempCapturer.CurrentAudioDevice;
            }

            for (int i = 0; i < _tempCapturer.AudioCodecsCount; i++)
            {

```

```

        string codec = _tempCapturer.GetAudioCodecName(i);
        cmbAviAudioCodecs.Items.Add(codec);
    }

    for (int i = 0; i < cmbAviAudioCodecs.Items.Count; i++)
    {
        if (cmbAviAudioCodecs.Items[i].ToString() == Program.Cfg.AviAudioCodec)
        {
            cmbAviAudioCodecs.SelectedIndex = i;
            break;
        }
    }

    if (cmbAviAudioCodecs.SelectedIndex == -1)
    {
        cmbAviAudioCodecs.SelectedItem = _tempCapturer.CurrentAudioCodec;
        Program.Cfg.AviAudioCodec = _tempCapturer.CurrentAudioCodec;
    }

    if (_tempCapturer.WebCamCount > 0)
    {

        for (int i = 0; i < _tempCapturer.WebCamCount; i++)
        {
            string camera = _tempCapturer.GetWebCamName(i);
            cmbWebCameras.Items.Add(camera);
        }

        for (int i = 0; i < cmbWebCameras.Items.Count; i++)
        {
            if (cmbWebCameras.Items[i].ToString() == Program.Cfg.WebCameraDevice)
            {
                cmbWebCameras.SelectedIndex = i;
                break;
            }
        }

        if (cmbWebCameras.SelectedIndex == -1 && cmbWebCameras.Items.Count > 0)
        {
            cmbWebCameras.SelectedIndex = _tempCapturer.CurrentWebCam;
            Program.Cfg.WebCameraDevice = _tempCapturer.CurrentWebCamName;
        }
    }
    else
    {
        cmbWebCameras.Enabled = false;
        tbWebCameraHeight.Enabled = false;
        tbWebCameraWidth.Enabled = false;
        tbWebCameraX.Enabled = false;
        tbWebCameraY.Enabled = false;
    }

    for (int i = 0; i < _tempCapturer.VideoCodecsCount; i++)
    {
        string codec = _tempCapturer.GetVideoCodecName(i);
        cmbAviVideoCodecs.Items.Add(codec);
    }

    for (int i = 0; i < cmbAviVideoCodecs.Items.Count; i++)
    {

```

```

                if (cmbAviVideoCodecs.Items[i].ToString() == Program.Cfg.AviVideoCodec)
                {
                    cmbAviVideoCodecs.SelectedIndex = i;
                    break;
                }
            }

            if (cmbAviVideoCodecs.SelectedIndex == -1)
            {
                cmbAviVideoCodecs.SelectedItem = _tempCapturer.CurrentAviVideoCodec;
                Program.Cfg.AviVideoCodec = _tempCapturer.CurrentVideoCodecName;
            }

            for (int i = 0; i < _tempCapturer.WMVAudioCodecsCount; i++)
            {
                string codec = _tempCapturer.GetWMVAudioCodecName(i);
                cmbWmvAudioCodecs.Items.Add(codec);
            }

            for (int i = 0; i < cmbWmvAudioCodecs.Items.Count; i++)
            {
                if (cmbWmvAudioCodecs.Items[i].ToString() == Program.Cfg.WmvAudioCodec)
                {
                    cmbWmvAudioCodecs.SelectedIndex = i;
                    break;
                }
            }

            if (cmbWmvAudioCodecs.SelectedIndex == -1)
            {
                cmbWmvAudioCodecs.SelectedItem = _tempCapturer.CurrentWmvAudioCodec;
                Program.Cfg.WmvAudioCodec = _tempCapturer.CurrentWMVAudioCodec;
            }

            for (int i = 0; i < _tempCapturer.WMVVideoCodecsCount; i++)
            {
                string codec = _tempCapturer.GetWMVVideoCodecName(i);
                cmbWmvVideoCodecs.Items.Add(codec);
            }

            for (int i = 0; i < cmbWmvVideoCodecs.Items.Count; i++)
            {
                if (cmbWmvVideoCodecs.Items[i].ToString() == Program.Cfg.WmvVideoCodec)
                {
                    cmbWmvVideoCodecs.SelectedIndex = i;
                    break;
                }
            }

            if (cmbWmvVideoCodecs.SelectedIndex == -1)
            {
                cmbWmvVideoCodecs.SelectedItem = _tempCapturer.CurrentWmvVideoCodec;
                Program.Cfg.WmvVideoCodec = _tempCapturer.CurrentWMVVideoCodec;
            }

            cbResizeVideo.Checked = Program.Cfg.ResizeOutputVideo;
            tbWidth.Text = Program.Cfg.OutputWidth.ToString();
            tbHeight.Text = Program.Cfg.OutputHeight.ToString();
            cbKeepAspectRatio.Checked = Program.Cfg.KeepAspectRatio;
            cmbFPS.Text = Program.Cfg.FPS.ToString();
        }
    }
}

```

```

cbLog.Checked = Program.Cfg.WriteLog;

tbWebCameraX.Text = Program.Cfg.WebCameraWindowX.ToString();
tbWebCameraY.Text = Program.Cfg.WebCameraWindowY.ToString();
tbWebCameraWidth.Text = Program.Cfg.WebCameraWindowWidth.ToString();
tbWebCameraHeight.Text = Program.Cfg.WebCameraWindowHeight.ToString();

cmbAudioLines.Enabled = cbEnableAudio.Checked && _tempCapturer.AudioDeviceCount > 0;
tbWidth.Enabled = cbResizeVideo.Checked;
tbHeight.Enabled = cbResizeVideo.Checked;
cbKeepAspectRatio.Enabled = cbResizeVideo.Checked;

tabControl2.SelectedIndex = Program.Cfg.SelectedVideoCodecTab;

cbCaptureMouseCursor.Checked = Program.Cfg.CaptureMouseCursor;
cbShowMouseHotSpot.Checked = Program.Cfg.ShowMouseHotSpot;
tbMouseAnimationDuration.Text = Program.Cfg.MouseAnimationDuration.ToString();
tbHotSpotRadius.Text = Program.Cfg.MouseSpotRadius.ToString(Culture);
ccMouseHotSpotColor.ForeColor = Program.Cfg.MouseHotSpotColor;
cbAnimateMouseClicks.Checked = Program.Cfg.AnimateMouseClicks;
cbAnimateMouseButtons.Checked = Program.Cfg.AnimateMouseButtons;
ccMouseCursorLeftClickAnimationColor.ForeColor = Program.Cfg.MouseCursorLeftClickAnimationColor;
ccMouseCursorRightClickAnimationColor.ForeColor = Program.Cfg.MouseCursorRightClickAnimationColor;

cmbFrameType.Items.Add("None");
cmbFrameType.Items.Add("Solid");
cmbFrameType.Items.Add("Dashed");
cmbFrameType.Items.Add("Dotted");
cmbFrameType.SelectedIndex = (int) Program.Cfg.CaptureAreaBorderType;
ccFrameColor.ForeColor = Program.Cfg.CaptureAreaBorderColor;
tbFrameWidth.Text = Program.Cfg.CaptureAreaBorderWidth.ToString(CultureInfo.InvariantCulture));
}

private void cbEnableAudio_CheckedChanged(object sender, EventArgs e)
{
    cmbAudioDevices.Enabled = cbEnableAudio.Checked;
    cmbAudioLines.Enabled = cbEnableAudio.Checked && _tempCapturer.AudioDeviceCount > 0;
}

private void cbResizeVideo_CheckedChanged(object sender, EventArgs e)
{
    tbWidth.Enabled = cbResizeVideo.Checked;
    tbHeight.Enabled = cbResizeVideo.Checked;
    cbKeepAspectRatio.Enabled = cbResizeVideo.Checked;
}

private void btnOk_Click(object sender, EventArgs e)
{
    if (!cbEnableAudio.Checked && _tempCapturer.AudioDeviceCount > 0)
        _tempCapturer.CurrentAudioDeviceLineCount > 0 && cmbAudioLines.SelectedIndex > -1)
    {
        MessageBox.Show("Please select Audio Line", Application.ProductName);
        return;
    }

    if (_tempCapturer.WMVAudioCodecsCount > 0 && cmbWmvAudioFormats.SelectedIndex > -1)
    {
        MessageBox.Show("Please select WMV Audio Format", Application.ProductName);
        return;
    }
}

```

```

        try
        {
            Program.Cfg.SelectedVideoCodecTab = tabControl2.SelectedItem;
            Program.Cfg.AudioDevice = cmbAudioDevices.SelectedItem;
            Program.Cfg.AudioLine = cmbAudioLines.SelectedItem != null ? cmbAudioLines.SelectedItem : null;
            Program.Cfg.EnableAudio = cbEnableAudio.Checked;
            Program.Cfg.AviAudioCodec = cmbAviAudioCodecs.SelectedItem.ToString();
            Program.Cfg.AviVideoCodec = cmbAviVideoCodecs.SelectedItem.ToString();
            Program.Cfg.WmvAudioCodec = cmbWmvAudioCodecs.SelectedItem.ToString();
            Program.Cfg.WmvAudioFormat = cmbWmvAudioFormats.SelectedItem.ToString();
            Program.Cfg.WmvVideoCodec = cmbWmvVideoCodecs.SelectedItem.ToString();
            Program.Cfg.ResizeOutputVideo = cbResizeVideo.Checked;
            Program.Cfg.OutputWidth = Int32.Parse(tbWidth.Text);
            Program.Cfg.OutputHeight = Int32.Parse(tbHeight.Text);
            Program.Cfg.KeepAspectRatio = cbKeepAspectRatio.Checked;
            Program.Cfg.FPS = float.Parse(cmbFPS.Text);
            Program.Cfg.WriteLog = cbLog.Checked;
        }
        if (cmbWebCameras.Enabled)
        {
            Program.Cfg.WebCameraDevice = cmbWebCameras.SelectedItem.ToString();
            Program.Cfg.WebCameraWindowX = Int32.Parse(tbWebCameraX.Text);
            Program.Cfg.WebCameraWindowY = Int32.Parse(tbWebCameraY.Text);
            Program.Cfg.WebCameraWindowWidth = Int32.Parse(tbWebCameraWidth.Text);
            Program.Cfg.WebCameraWindowHeight = Int32.Parse(tbWebCameraHeight.Text);
        }
        Program.Cfg.CaptureMouseCursor = cbCaptureMouseCursor.Checked;
        Program.Cfg.ShowMouseHotSpot = cbShowMouseHotSpot.Checked;
        Program.Cfg.MouseAnimationDuration = Convert.ToInt32(tbMouseDuration.Text);
        Program.Cfg.MouseSpotRadius = Convert.ToInt32(tbHotSpotRadius.Text);
        Program.Cfg.MouseHotSpotColor = ccMouseHotSpotColor.ForeColor;
        Program.Cfg.AnimateMouseClicks = cbAnimateMouseClicks.Checked;
        Program.Cfg.AnimateMouseButtons = cbAnimateMouseButtons.Checked;
        Program.Cfg.MouseCursorLeftClickAnimationColor = ccMouseLeftClickAnimationColor;
        Program.Cfg.MouseCursorRightClickAnimationColor = ccMouseRightClickAnimationColor;
        Program.Cfg.CaptureAreaBorderType = (CaptureAreaBorderType)cbCaptureAreaBorderType.SelectedIndex;
        Program.Cfg.CaptureAreaBorderColor = ccFrameColor.ForeColor;
        Program.Cfg.CaptureAreaBorderWidth = Int32.Parse(tbFrameWidth.Text);
    }
    catch (Exception exception)
    {
        MessageBox.Show(exception.Message, Application.ProductName);
    }
}

private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    Process prc = new Process();
    prc.StartInfo.FileName = "http://www.bytescout.com/";
    prc.StartInfo.UseShellExecute = true;

    try
    {
        prc.Start();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, Application.ProductName);
    }
}

```

```

private void btnVideoCodecProps_Click(object sender, EventArgs e)
{
    Capturer tempCapturer = new Capturer();
    tempCapturer.RegistrationName = "demo";
    tempCapturer.RegistrationKey = "demo";

    tempCapturer.CurrentVideoCodecName = cmbAviVideoCodecs.SelectedItem.ToString();

    try
    {
        tempCapturer.ShowVideoCodecSettingsDialog(0);
    }
    catch (InvalidCastException)
    {
        MessageBox.Show("This codec has no properties dialog.");
    }
    catch (Exception exception)
    {
        MessageBox.Show("Failed to open the codec properties dialog.");
    }
}

finally
{
    Marshal.ReleaseComObject(tempCapturer);
}
}

private void linkViewLog_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    string logFile = Path.GetTempPath() + Application.ProductName + ".log";

    if (File.Exists(logFile))
    {
        Process prc = new Process();
        prc.StartInfo.FileName = logFile;
        prc.StartInfo.UseShellExecute = true;

        try
        {
            prc.Start();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, Application.ProductName);
        }
    }
}

private void cmbAudioDevices_SelectedIndexChanged(object sender, EventArgs e)
{
    if (_tempCapturer.AudioDeviceCount == 0)
        return;

    _tempCapturer.CurrentAudioDeviceName = cmbAudioDevices.SelectedItem.ToString();

    cmbAudioLines.Items.Clear();

    for (int i = 0; i < _tempCapturer.CurrentAudioDeviceLineCount;
    {

```

```

        string line = _tempCapturer.GetCurrentAudioDeviceLineName();
        cmbAudioLines.Items.Add(line);
    }

    for (int i = 0; i < cmbAudioLines.Items.Count; i++)
    {
        if (cmbAudioLines.Items[i].ToString() == Program.Cfg.AudioLine)
        {
            cmbAudioLines.SelectedIndex = i;
            break;
        }
    }

    if (cmbAudioLines.SelectedIndex == -1)
    {
        for (int j = 0; j < cmbAudioLines.Items.Count; j++ )
        {
            string tmpS = cmbAudioLines.Items[j].ToString().ToUpper();
            if (tmpS.IndexOf("MIC") > -1)
            {
                cmbAudioLines.SelectedIndex = j;
            }
        }
    }

    if (cmbAudioLines.SelectedIndex == -1)
    {
        cmbAudioLines.SelectedItem = _tempCapturer.CurrentAudioDeviceLineName;
        Program.Cfg.AudioLine = _tempCapturer.CurrentAudioDeviceLineName;
    }
}

private void cmbWmvAudioCodecs_SelectedIndexChanged(object sender, EventArgs e)
{
    _tempCapturer.CurrentWMVAudioCodecName = cmbWmvAudioCodecs.SelectedItem.ToString();

    cmbWmvAudioFormats.Items.Clear();

    // Get list of available WMV audio formats
    for (int i = 0; i < _tempCapturer.WMVFormatCount; i++)
    {
        string codec = _tempCapturer.GetWMVFormatDescription(i);
        cmbWmvAudioFormats.Items.Add(codec);
    }

    for (int i = 0; i < cmbWmvAudioFormats.Items.Count; i++)
    {
        if (i == Program.Cfg.WmvAudioFormat)
        {
            cmbWmvAudioFormats.SelectedIndex = i;
            break;
        }
    }

    if (cmbWmvAudioFormats.SelectedIndex == -1)
    {
        cmbWmvAudioFormats.SelectedIndex = _tempCapturer.CurrentWMVFormatIndex;
        Program.Cfg.WmvAudioFormat = _tempCapturer.CurrentWMVFormatName;
    }
}

```

```
    }  
}
```

VIDEO

<https://www.youtube.com/watch?v=fujkvtWUVCw>

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout Screen Capturing SDK Home Page](#)
[Explore ByteScout Screen Capturing SDK Documentation](#)
[Explore Samples](#)
[Sign Up for ByteScout Screen Capturing SDK Online Training](#)

ON-DEMAND REST WEB API

[Get Your API Key](#)
[Explore Web API Docs](#)
[Explore Web API Samples](#)

[visit www.ByteScout.com](#)

[visit www.PDF.co](#)

[www.bytescout.com](#)