

How to read barcode from live video cam in C# using ByteScout BarCode Reader SDK

The tutorial shows how to read barcode from live video cam in C#

Read barcode from live video cam is easy to implement in C# if you use these source codes below. ByteScout BarCode Reader SDK is the SDK for barcode decoding. Can read all popular types from Code 128, GS1, UPC and Code 39 to QR Code, Datamatrix, PDF417. Images, pdf, TIF images and live web camera are supported as input. Designed to handle documents with noise and defects. Includes optional splitter and merger for pdf and tiff based on barcodes. Batch mode is optimized for high performance with multiple threads. Decoded values can be exported to XML, JSON, CSV or into custom data format. It can be used to read barcode from live video cam using C#.

You will save a lot of time on writing and testing code as you may just take the C# code from ByteScout BarCode Reader SDK for read barcode from live video cam below and use it in your application. This C# sample code is all you need for your app. Just copy and paste the code, add references (if needs to) and you are all set! Enjoy writing a code with ready-to-use sample C# codes.

Trial version of ByteScout BarCode Reader SDK can be downloaded for free from our website. It also includes source code samples for C# and other programming languages.

MainForm.Designer.cs

```
namespace ReadFromVideoCamera
{
    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; otherwise, false; otherwise, false if disposing is true.
        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.lblFoundBarcodes = new System.Windows.Forms.Label();
    this.pictureBoxPreview = new System.Windows.Forms.PictureBox();
    this.btnStop = new System.Windows.Forms.Button();
    this.btnStart = new System.Windows.Forms.Button();
    this.rtbFoundBarcodes = new System.Windows.Forms.RichTextBox();
    this.label1 = new System.Windows.Forms.Label();
    this.cmbCamera = new System.Windows.Forms.ComboBox();
    this.btnExit = new System.Windows.Forms.Button();
    this.label3 = new System.Windows.Forms.Label();
    this.cmbBarcodeType = new System.Windows.Forms.ComboBox();
    this.lblScanning = new System.Windows.Forms.Label();
    this.label2 = new System.Windows.Forms.Label();
    this.tbCameraWidth = new System.Windows.Forms.TextBox();
    this.tbCameraHeight = new System.Windows.Forms.TextBox();
    this.label4 = new System.Windows.Forms.Label();
    this.cbStopOnFirstBarcode = new System.Windows.Forms.CheckBox();
    this.btnUpdateCameraImageDimensions = new System.Windows.Forms.Button();
    this.btnCameraProperties = new System.Windows.Forms.Button();
    ((System.ComponentModel.ISupportInitialize)(this.pictureBoxPreview)).BeginInit();
    this.SuspendLayout();
    //
    // lblFoundBarcodes
    //
    this.lblFoundBarcodes.Anchor = ((System.Windows.Forms.AnchorStyles)
    | System.Windows.Forms.AnchorStyles.Left)
    | System.Windows.Forms.AnchorStyles.Right));
    this.lblFoundBarcodes.AutoSize = true;
    this.lblFoundBarcodes.Location = new System.Drawing.Point(12, 6);
    this.lblFoundBarcodes.Name = "lblFoundBarcodes";
    this.lblFoundBarcodes.Size = new System.Drawing.Size(93, 13);
    this.lblFoundBarcodes.TabIndex = 5;
    this.lblFoundBarcodes.Text = "Found 0 barcodes";
    //
    // pictureBoxPreview
    //
    this.pictureBoxPreview.Anchor = ((System.Windows.Forms.AnchorStyles)
    | System.Windows.Forms.AnchorStyles.Left)
    | System.Windows.Forms.AnchorStyles.Right));
    this.pictureBoxPreview.BackColor = System.Drawing.Color.White;
    this.pictureBoxPreview.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
    this.pictureBoxPreview.Location = new System.Drawing.Point(12, 19);
    this.pictureBoxPreview.Name = "pictureBoxPreview";
    this.pictureBoxPreview.Size = new System.Drawing.Size(640, 480);
    this.pictureBoxPreview.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;
    this.pictureBoxPreview.TabIndex = 8;
    this.pictureBoxPreview.TabStop = false;
    //
    // btnStop
    //
    this.btnStop.Anchor = ((System.Windows.Forms.AnchorStyles)
    | System.Windows.Forms.AnchorStyles.Right));
    this.btnStop.Enabled = false;
    this.btnStop.Font = new System.Drawing.Font("Tahoma", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));
    this.btnStop.Location = new System.Drawing.Point(825, 19);
    this.btnStop.Name = "btnStop";
    this.btnStop.Size = new System.Drawing.Size(75, 23);
    this.btnStop.TabIndex = 9;
    this.btnStop.Text = "Stop";
    this.btnStop.UseVisualStyleBackColor = true;
}

#endregion

```

```

this.btnStop.Location = new System.Drawing.Point(98, 690);
this.btnStop.Name = "btnStop";
this.btnStop.Size = new System.Drawing.Size(80, 26);
this.btnStop.TabIndex = 7;
this.btnStop.Text = "Stop";
this.btnStop.UseVisualStyleBackColor = true;
this.btnStop.Click += new System.EventHandler(this.btnStop_Click)
//
// btnStart
//
this.btnStart.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.btnStart.Font = new System.Drawing.Font("Tahoma", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));
this.btnStart.Location = new System.Drawing.Point(12, 690);
this.btnStart.Name = "btnStart";
this.btnStart.Size = new System.Drawing.Size(80, 26);
this.btnStart.TabIndex = 6;
this.btnStart.Text = "Start";
this.btnStart.UseVisualStyleBackColor = true;
this.btnStart.Click += new System.EventHandler(this.btnStart_Click)
//
// rtbFoundBarcodes
//
this.rtbFoundBarcodes.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.rtbFoundBarcodes.Location = new System.Drawing.Point(12, 690);
this.rtbFoundBarcodes.Name = "rtbFoundBarcodes";
this.rtbFoundBarcodes.Size = new System.Drawing.Size(640, 68);
this.rtbFoundBarcodes.TabIndex = 5;
this.rtbFoundBarcodes.Text = "";
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(9, 42);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(80, 13);
this.label1.TabIndex = 20;
this.label1.Text = "Camera Device";
//
// cmbCamera
//
this.cmbCamera.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmbCamera.FormattingEnabled = true;
this.cmbCamera.Location = new System.Drawing.Point(163, 39);
this.cmbCamera.Name = "cmbCamera";
this.cmbCamera.Size = new System.Drawing.Size(245, 21);
this.cmbCamera.TabIndex = 1;
this.cmbCamera.SelectedIndexChanged += new System.EventHandler(this.cmbCamera_SelectedIndexChanged)
//
// btnExit
//
this.btnExit.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.btnExit.DialogResult = System.Windows.Forms.DialogResult.Cancel;
this.btnExit.Font = new System.Drawing.Font("Tahoma", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));
this.btnExit.Location = new System.Drawing.Point(562, 690);
this.btnExit.Name = "btnExit";
this.btnExit.Size = new System.Drawing.Size(90, 26);
this.btnExit.TabIndex = 8;
this.btnExit.Text = "Exit";
this.btnExit.UseVisualStyleBackColor = true;

```

```

this.btnExit.Click += new System.EventHandler(this.btnExit_Click)
//
// label3
//
this.label3.AutoSize = true;
this.label3.Font = new System.Drawing.Font("Tahoma", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));
this.label3.Location = new System.Drawing.Point(9, 15);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(131, 13);
this.label3.TabIndex = 23;
this.label3.Text = "Barcode Type To Scan";
//
// cmbBarcodeType
//
this.cmbBarcodeType.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmbBarcodeType.FormattingEnabled = true;
this.cmbBarcodeType.Location = new System.Drawing.Point(163, 11);
this.cmbBarcodeType.Name = "cmbBarcodeType";
this.cmbBarcodeType.Size = new System.Drawing.Size(245, 21);
this.cmbBarcodeType.Sorted = true;
this.cmbBarcodeType.TabIndex = 0;
//
// lblScanning
//
this.lblScanning.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Right)));
this.lblScanning.AutoSize = true;
this.lblScanning.Font = new System.Drawing.Font("Tahoma", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)0));
this.lblScanning.Location = new System.Drawing.Point(580, 600);
this.lblScanning.Name = "lblScanning";
this.lblScanning.Size = new System.Drawing.Size(67, 13);
this.lblScanning.TabIndex = 24;
this.lblScanning.Text = "Scanning...";
this.lblScanning.Visible = false;
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(9, 69);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(132, 13);
this.label2.TabIndex = 26;
this.label2.Text = "Camera Image Dimensions";
//
// tbCameraWidth
//
this.tbCameraWidth.Location = new System.Drawing.Point(163, 66);
this.tbCameraWidth.Name = "tbCameraWidth";
this.tbCameraWidth.Size = new System.Drawing.Size(35, 20);
this.tbCameraWidth.TabIndex = 2;
this.tbCameraWidth.Text = "640";
//
// tbCameraHeight
//
this.tbCameraHeight.Location = new System.Drawing.Point(222, 66);
this.tbCameraHeight.Name = "tbCameraHeight";
this.tbCameraHeight.Size = new System.Drawing.Size(35, 20);
this.tbCameraHeight.TabIndex = 3;
this.tbCameraHeight.Text = "480";
//
// label4

```

```

//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(204, 69);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(12, 13);
this.label4.TabIndex = 29;
this.label4.Text = "x";
//
// cbStopOnFirstBarcode
//
this.cbStopOnFirstBarcode.AutoSize = true;
this.cbStopOnFirstBarcode.Location = new System.Drawing.Point(
this.cbStopOnFirstBarcode.Name = "cbStopOnFirstBarcode";
this.cbStopOnFirstBarcode.Size = new System.Drawing.Size(154, 1
this.cbStopOnFirstBarcode.TabIndex = 30;
this.cbStopOnFirstBarcode.Text = "Stop on first barcode found"
this.cbStopOnFirstBarcode.UseVisualStyleBackColor = true;
//
// btnUpdateCameraImageDimensions
//
this.btnUpdateCameraImageDimensions.Location = new System.Draw
this.btnUpdateCameraImageDimensions.Name = "btnUpdateCameraIma
this.btnUpdateCameraImageDimensions.Size = new System.Drawing.
this.btnUpdateCameraImageDimensions.TabIndex = 4;
this.btnUpdateCameraImageDimensions.Text = "Update";
this.btnUpdateCameraImageDimensions.UseVisualStyleBackColor =
this.btnUpdateCameraImageDimensions.Click += new System.EventH
//
// btnCameraProperties
//
this.btnCameraProperties.Location = new System.Drawing.Point(4
this.btnCameraProperties.Name = "btnCameraProperties";
this.btnCameraProperties.Size = new System.Drawing.Size(131, 2
this.btnCameraProperties.TabIndex = 31;
this.btnCameraProperties.Text = "Camera Properties";
this.btnCameraProperties.UseVisualStyleBackColor = true;
this.btnCameraProperties.Click += new System.EventHandler(this
//
// MainForm
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.CancelButton = this.btnExit;
this.ClientSize = new System.Drawing.Size(664, 728);
this.Controls.Add(this.btnCameraProperties);
this.Controls.Add(this.btnUpdateCameraImageDimensions);
this.Controls.Add(this.cbStopOnFirstBarcode);
this.Controls.Add(this.label4);
this.Controls.Add(this.tbCameraHeight);
this.Controls.Add(this.tbCameraWidth);
this.Controls.Add(this.label2);
this.Controls.Add(this.lblScanning);
this.Controls.Add(this.label3);
this.Controls.Add(this.cmbBarcodeType);
this.Controls.Add(this.btnExit);
this.Controls.Add(this.label1);
this.Controls.Add(this.cmbCamera);
this.Controls.Add(this.rtbFoundBarcodes);
this.Controls.Add(this.btnStop);
this.Controls.Add(this.btnStart);

```

```

        this.Controls.Add(this.pictureBoxPreview);
        this.Controls.Add(this.lblFoundBarcodes);
        this.Name = "MainForm";
        this.ShowIcon = false;
        this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;
        this.Text = "Barcode Scanner";
        this.Load += new System.EventHandler(this.Form_Load);
        ((System.ComponentModel.ISupportInitialize)(this.pictureBoxPreview)).BeginInit();
        this.ResumeLayout(false);
        this.PerformLayout();

    }

    #endregion

    private System.Windows.Forms.Label lblFoundBarcodes;
    private System.Windows.Forms.PictureBox pictureBoxPreview;
    private System.Windows.Forms.Button btnStop;
    private System.Windows.Forms.Button btnStart;
    private System.Windows.Forms.RichTextBox rtbFoundBarcodes;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.ComboBox cmbCamera;
    private System.Windows.Forms.Button btnExit;
    private System.Windows.Forms.Label label3;
    private System.Windows.Forms.ComboBox cmbBarcodeType;
    private System.Windows.Forms.Label lblScanning;
    private System.Windows.Forms.Label label2;
    private System.Windows.Forms.TextBox tbCameraWidth;
    private System.Windows.Forms.TextBox tbCameraHeight;
    private System.Windows.Forms.Label label4;
        private System.Windows.Forms.CheckBox cbStopOnFirstBarcode;
    private System.Windows.Forms.Button btnUpdateCameraImageDimensions;
    private System.Windows.Forms.Button btnCameraProperties;
}

```

MainForm.cs

```

using System;
using System.ComponentModel;
using System.Drawing;
using System.Media;
using System.Reflection;
using System.Text;
using System.Windows.Forms;
using System.Threading;
using Bytescout.BarCodeReader;
using TouchlessLib;

```

```

namespace ReadFromVideoCamera
{
    public partial class MainForm: Form
    {
        // Scan delay, ms.
        const int SCAN_DELAY = 1500; // scan barcodes every 1.5 sec

        // Touchless SDK library manager (to use it you should have TouchlessL
        readonly TouchlessMgr _touchlessLibManager;

        // Background thread for barcode scanning
        readonly BackgroundWorker _backgroundWorker = new BackgroundWorker();
        // Synchronization event
        readonly AutoResetEvent _synchronizationEvent = new AutoResetEvent(false);

        // Form constructor
        public MainForm()
        {
            InitializeComponent();

            // Create Touchless library manager
            _touchlessLibManager = new TouchlessMgr();

            // Setup background worker
            _backgroundWorker.WorkerSupportsCancellation = true;
            _backgroundWorker.DoWork += BackgroundWorker_DoWork;
            _backgroundWorker.RunWorkerCompleted += BackgroundWorker_RunWo
        }

        // On form loading
        private void Form_Load(object sender, EventArgs e)
        {
            // Fill devices combobox with available video cameras
            foreach (Camera camera in _touchlessLibManager.Cameras)
                cmbCamera.Items.Add(camera);

            // Select the first available camera. See also cmbCamera_SelectedIndexChang
            if (_touchlessLibManager.Cameras.Count > 0)
                cmbCamera.SelectedItem = _touchlessLibManager.Cameras[0];
            else
                MessageBox.Show("No video camera available. Please connect the

            // Populate barcode types combobox
            PopulateBarcodeTypesCombobox();
            // Select some default barcode type
            cmbBarcodeType.SelectedItem = "QRCode";
        }

        protected void PopulateBarcodeTypesCombobox()
        {
            cmbBarcodeType.Items.Clear();

            foreach (PropertyInfo propertyInfo in typeof(BarcodeTypeSelector).C
                cmbBarcodeType.Items.Add(propertyInfo.Name);
        }

        // On camera selected
        private void cmbCamera_SelectedIndexChanged(object sender, EventArgs e)
        {

```

```

if (_touchlessLibManager.CurrentCamera != null)
    _touchlessLibManager.CurrentCamera.OnImageCaptured -= CurrentCamera_OnImageCaptured;

    if (cmbCamera.SelectedIndex != -1)
    {
        Camera camera = _touchlessLibManager.Cameras[cmbCamera.SelectedIndex];

        if (camera != null)
        {
            // Set camera output image dimensions
            camera.CaptureWidth = int.Parse(tbCameraWidth.Text);
            camera.CaptureHeight = int.Parse(tbCameraHeight.Text);

            camera.OnImageCaptured += CurrentCamera_OnImageCaptured;

            // Select the camera
            _touchlessLibManager.CurrentCamera = camera;
        }
    }
}

private void btnUpdateCameraImageDimensions_Click(object sender, EventArgs e)
{
    if (_touchlessLibManager.CurrentCamera != null)
    {
        // Update camera's output image dimensions
        _touchlessLibManager.CurrentCamera.CaptureWidth = int.Parse(tbCameraWidth.Text);
        _touchlessLibManager.CurrentCamera.CaptureHeight = int.Parse(tbCameraHeight.Text);
    }
}

public void StartDecoding()
{
    if (cmbCamera.SelectedIndex == -1)
        return;

    // Clear the output text box
    rtbFoundBarcodes.Clear();

    // Check if we have camera selected
    if (cmbCamera.SelectedIndex != -1)
    {
        // Start the decoding in the background thread
        BarcodeTypeSelector barcodeTypesToFind = GetBarcodeTypeFromComboBox(cmbBarcodeTypes);
        _backgroundWorker.RunWorkerAsync(barcodeTypesToFind);

        UpdateControls(true);
    }
    else
    {
        MessageBox.Show("Please select the camera first!");
    }
}

private void StopDecoding()
{
    _backgroundWorker.CancelAsync();

    // Wait until BackgroundWorker finished
    if (_backgroundWorker.IsBusy)

```



```

        _synchronizationEvent.WaitOne();

        UpdateControls(false);
    }

    void UpdateControls(bool started)
    {
        if (started)
        {
            btnStart.Enabled = false;
            btnStop.Enabled = true;
            cmbBarcodeType.Enabled = false;
            cmbCamera.Enabled = false;
            tbCameraHeight.Enabled = false;
            tbCameraWidth.Enabled = false;
            btnUpdateCameraImageDimensions.Enabled = false;
            cbStopOnFirstBarcode.Enabled = false;
            lblScanning.Visible = true;
            lblScanning.Text = "Scanning...";
        }
        else
        {
            btnStart.Enabled = true;
            btnStop.Enabled = false;
            cmbBarcodeType.Enabled = true;
            cmbCamera.Enabled = true;
            cbStopOnFirstBarcode.Enabled = true;
            tbCameraHeight.Enabled = true;
            tbCameraWidth.Enabled = true;
            btnUpdateCameraImageDimensions.Enabled = true;
            lblScanning.Visible = true;
        }
    }

    void CurrentCamera_OnImageCaptured(object sender, CameraEventArgs e)
    {
        pictureBoxPreview.Image = e.Image;
    }

    private void btnStart_Click(object sender, EventArgs e)
    {
        StartDecoding();
    }

    private void btnStop_Click(object sender, EventArgs e)
    {
        StopDecoding();
    }

    // Background thread procedure used by BackgroundWorker
    public void BackgroundWorker_DoWork(object sender, DoWorkEventArgs e)
    {
        BackgroundWorker worker = (BackgroundWorker) sender;
        BarcodeTypeSelector barcodeTypesToFind = (BarcodeTypeSelector)

        // Create and setup barcode reader instance
        using (Reader reader = new Reader())
        {
            reader.RegistrationName = "demo";
            reader.RegistrationKey = "demo";
        }
    }

```

```

        reader.BarcodeTypesToFind = barcodeTypesToFind;

        // Work while not canceled
        while (true)
        {
            // Check cancellation
            if (worker.CancellationPending)
            {
                e.Cancel = true;
                _synchronizationEvent.Set();
                return;
            }

            // Get image from camera by invoking method from
            Bitmap bitmap = (Bitmap) Invoke(new GetCameraImageDelegate());
            if (bitmap == null)
            {
                e.Result = null;
                return;
            }

            // Search the image for barcodes
            FoundBarcode[] result = reader.ReadFrom(bitmap);

            // Update UI asynchronously
            BeginInvoke(new Action<FoundBarcode[]>(UpdateStatus), result);

            // Pause
            Thread.Sleep(SCAN_DELAY);
        }
    }

    delegate Bitmap GetCameraImageDelegate();

    Bitmap GetCameraImage()
    {
        if (!IsDisposed && !Disposing && _touchlessLibManager.CurrentCamera != null)
            return _touchlessLibManager.CurrentCamera.GetCurrentImage();

        return null;
    }

    // Update UI with found barcodes information
    void UpdateStatus(FoundBarcode[] foundBarcodes)
    {
        if (foundBarcodes != null && foundBarcodes.Length > 0)
        {
            // Play sound if we found any barcode
            SystemSounds.Beep.Play();

            StringBuilder stringBuilder = new StringBuilder();

            stringBuilder.AppendFormat("Time: {0:HH:mm:ss:tt}", DateTime.Now);
            stringBuilder.AppendLine();

            // Display found barcodes in the output text box
            foreach (FoundBarcode barcode in foundBarcodes)
            {

```

```

        stringBuilder.AppendFormat("Found barcode: {0}\n");
        stringBuilder.AppendLine();
    }

    rtbFoundBarcodes.Text = stringBuilder.ToString();

    // Update status text with number of found barcodes
    lblFoundBarcodes.Text = string.Format("Found {0} barcodes");
}

// Make "Scanning..." label flicker.
lblScanning.Visible = !lblScanning.Visible;
lblScanning.Refresh();

// Check if we need to stop on first barcode found
if (cbStopOnFirstBarcode.Checked && foundBarcodes != null && foundBarcodes.Count > 0)
{
    StopDecoding();
}
}

// Background thread is finished
private void BackgroundWorker_RunWorkerCompleted(object sender, RunWorkerCompletedEventArgs e)
{
    // Update UI asynchronously
    BeginInvoke(new Action<RunWorkerCompletedEventArgs>(OnBackgroundWorkerFinished), e);
}

void OnBackgroundWorkerFinished(RunWorkerCompletedEventArgs completedEventArgs)
{
    if (completedEventArgs.Cancelled)
    {
        lblScanning.Text = "Stopped";
    }
    else if (completedEventArgs.Error != null)
    {
        lblScanning.Text = "Error: " + completedEventArgs.Error.Message;
    }
    else
    {
        lblScanning.Text = "Done!";
    }

    UpdateControls(false);
}

// Gets selected barcode type
private BarcodeTypeSelector GetBarcodeTypeFromCombobox()
{
    BarcodeTypeSelector result = new BarcodeTypeSelector();

    string selectedBarcodeTypeName = (string) cmbBarcodeType.SelectedItem;

    PropertyInfo propertyInfo = typeof(BarcodeTypeSelector).GetProperty(selectedBarcodeTypeName);
    propertyInfo.SetValue(result, true, null);

    return result;
}

protected override void OnClosing(CancelEventArgs e)

```

```

        {
            StopDecoding();

            _touchlessLibManager.Dispose();

            base.OnClosing(e);
        }

private void btnExit_Click(object sender, EventArgs e)
{
    DialogResult = DialogResult.OK;

    Close();
}

private void btnCameraProperties_Click(object sender, EventArgs e)
{
    try
    {
        if (_touchlessLibManager.CurrentCamera != null)
            _touchlessLibManager.CurrentCamera.ShowProperties();
    }
    catch (Exception exception)
    {
        MessageBox.Show(exception.Message);
    }
}
}
}

```

Program.cs

```

using System;
using System.Windows.Forms;

namespace ReadFromVideoCamera
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [MTAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);

            MainForm dlg = new MainForm();
            dlg.ShowDialog();
        }
    }
}

```

```
}  
}
```

ReadFromVideoCamera.VS2005.csproj

```
<?xml version="1.0" encoding="utf-8"?>  
<Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">  
  <PropertyGroup>  
    <Configuration Condition="'$(Configuration)' == ''">Debug</Configuration>  
    <Platform Condition="'$(Platform)' == ''">x86</Platform>  
    <ProductVersion>8.0.50727</ProductVersion>  
    <SchemaVersion>2.0</SchemaVersion>  
    <ProjectGuid>{9EACD999-022B-495A-BCDF-BB82C2534522}</ProjectGuid>  
    <OutputType>WinExe</OutputType>  
    <RootNamespace>ReadFromVideoCamera</RootNamespace>  
    <AssemblyName>ReadFromVideoCamera</AssemblyName>  
    <FileUpgradeFlags>  
  </FileUpgradeFlags>  
    <UpgradeBackupLocation />  
    <PublishUrl>publish\</PublishUrl>  
    <Install>true</Install>  
    <InstallFrom>Disk</InstallFrom>  
    <UpdateEnabled>>false</UpdateEnabled>  
    <UpdateMode>Foreground</UpdateMode>  
    <UpdateInterval>7</UpdateInterval>  
    <UpdateIntervalUnits>Days</UpdateIntervalUnits>  
    <UpdatePeriodically>>false</UpdatePeriodically>  
    <UpdateRequired>>false</UpdateRequired>  
    <MapFileExtensions>true</MapFileExtensions>  
    <ApplicationRevision>0</ApplicationRevision>  
    <ApplicationVersion>1.0.0.%2a</ApplicationVersion>  
    <IsWebBootstrapper>>false</IsWebBootstrapper>  
    <UseApplicationTrust>>false</UseApplicationTrust>  
    <BootstrapperEnabled>true</BootstrapperEnabled>  
    <OldToolsVersion>3.5</OldToolsVersion>  
  </PropertyGroup>  
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|x86'">  
    <DebugSymbols>>true</DebugSymbols>  
    <OutputPath>bin\x86\Debug\</OutputPath>  
    <DefineConstants>DEBUG;TRACE</DefineConstants>  
    <DebugType>full</DebugType>  
    <PlatformTarget>x86</PlatformTarget>  
    <ErrorReport>prompt</ErrorReport>  
    <CodeAnalysisRuleSet>AllRules.ruleset</CodeAnalysisRuleSet>  
    <CodeAnalysisIgnoreBuiltInRuleSets>>false</CodeAnalysisIgnoreBuiltInRuleSets>  
    <CodeAnalysisIgnoreBuiltInRules>>false</CodeAnalysisIgnoreBuiltInRules>  
  </PropertyGroup>  
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Release|x86'">  
    <OutputPath>bin\x86\Release\</OutputPath>  
    <DefineConstants>TRACE</DefineConstants>
```

```

<Optimize>true</Optimize>
<DebugType>pdbonly</DebugType>
<PlatformTarget>x86</PlatformTarget>
<ErrorReport>prompt</ErrorReport>
<CodeAnalysisRuleSet>AllRules.ruleset</CodeAnalysisRuleSet>
<CodeAnalysisIgnoreBuiltInRuleSets>>false</CodeAnalysisIgnoreBuiltInRuleSets>
<CodeAnalysisIgnoreBuiltInRules>>false</CodeAnalysisIgnoreBuiltInRules>
</PropertyGroup>
<PropertyGroup>
  <StartupObject>ReadFromVideoCamera.Program</StartupObject>
</PropertyGroup>
<ItemGroup>
  <Reference Include="Bytescout.BarCodeReader, Version=8.20.0.1340, Culture=neutral,
    <SpecificVersion>False</SpecificVersion>
    <HintPath>C:\Program Files\Bytescout BarCode Reader SDK\net2.00\Bytescout.BarCode
  </Reference>
  <Reference Include="System" />
  <Reference Include="System.Data" />
  <Reference Include="System.Deployment" />
  <Reference Include="System.Drawing" />
  <Reference Include="System.Windows.Forms" />
  <Reference Include="System.Xml" />
  <Reference Include="TouchlessLib, Version=1.0.0.0, Culture=neutral, processorArchit
    <SpecificVersion>False</SpecificVersion>
  </Reference>
</ItemGroup>
<ItemGroup>
  <Compile Include="MainForm.cs">
    <SubType>Form</SubType>
  </Compile>
  <Compile Include="MainForm.Designer.cs">
    <DependentUpon>MainForm.cs</DependentUpon>
  </Compile>
  <Compile Include="Program.cs" />
</ItemGroup>
<ItemGroup>
</ItemGroup>
<ItemGroup>
  <Content Include="TouchlessLib.dll">
  </Content>
  <Content Include="WebCamLib.dll">
    <CopyToOutputDirectory>Always</CopyToOutputDirectory>
  </Content>
</ItemGroup>
<ItemGroup>
  <EmbeddedResource Include="MainForm.resx">
    <DependentUpon>MainForm.cs</DependentUpon>
  </EmbeddedResource>
</ItemGroup>
<Import Project="$(MSBuildBinPath)\Microsoft.CSharp.targets" />
<!-- To modify your build process, add your task inside one of the targets below and
  Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>

```

ReadFromVideoCamera.VS2008.csproj

```
<?xml version="1.0" encoding="utf-8"?>
<Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup>
    <Configuration Condition="'$(Configuration)' == ''">Debug</Configuration>
    <Platform Condition="'$(Platform)' == ''">x86</Platform>
    <ProductVersion>9.0.30729</ProductVersion>
    <SchemaVersion>2.0</SchemaVersion>
    <ProjectGuid>{9EACD999-022B-495A-BCDF-BB82C2534522}</ProjectGuid>
    <OutputType>WinExe</OutputType>
    <RootNamespace>ReadFromVideoCamera</RootNamespace>
    <AssemblyName>ReadFromVideoCamera</AssemblyName>
    <FileUpgradeFlags>
    </FileUpgradeFlags>
    <UpgradeBackupLocation />
    <PublishUrl>publish\</PublishUrl>
    <Install>true</Install>
    <InstallFrom>Disk</InstallFrom>
    <UpdateEnabled>>false</UpdateEnabled>
    <UpdateMode>Foreground</UpdateMode>
    <UpdateInterval>7</UpdateInterval>
    <UpdateIntervalUnits>Days</UpdateIntervalUnits>
    <UpdatePeriodically>>false</UpdatePeriodically>
    <UpdateRequired>>false</UpdateRequired>
    <MapFileExtensions>>true</MapFileExtensions>
    <ApplicationRevision>0</ApplicationRevision>
    <ApplicationVersion>1.0.0.%2a</ApplicationVersion>
    <IsWebBootstrapper>>false</IsWebBootstrapper>
    <UseApplicationTrust>>false</UseApplicationTrust>
    <BootstrapperEnabled>>true</BootstrapperEnabled>
    <OldToolsVersion>3.5</OldToolsVersion>
    <TargetFrameworkVersion>v3.5</TargetFrameworkVersion>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|x86'">
    <DebugSymbols>>true</DebugSymbols>
    <OutputPath>bin\x86\Debug\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <DebugType>full</DebugType>
    <PlatformTarget>x86</PlatformTarget>
    <ErrorReport>prompt</ErrorReport>
    <CodeAnalysisRuleSet>AllRules.ruleset</CodeAnalysisRuleSet>
    <CodeAnalysisIgnoreBuiltInRuleSets>>false</CodeAnalysisIgnoreBuiltInRuleSets>
    <CodeAnalysisIgnoreBuiltInRules>>false</CodeAnalysisIgnoreBuiltInRules>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Release|x86'">
    <OutputPath>bin\x86\Release\</OutputPath>
    <DefineConstants>TRACE</DefineConstants>
    <Optimize>>true</Optimize>
    <DebugType>pdbonly</DebugType>
    <PlatformTarget>x86</PlatformTarget>
    <ErrorReport>prompt</ErrorReport>
    <CodeAnalysisRuleSet>AllRules.ruleset</CodeAnalysisRuleSet>
  </PropertyGroup>

```

```

    <CodeAnalysisIgnoreBuiltInRuleSets>>false</CodeAnalysisIgnoreBuiltInRuleSets>
    <CodeAnalysisIgnoreBuiltInRules>>false</CodeAnalysisIgnoreBuiltInRules>
</PropertyGroup>
<PropertyGroup>
    <StartupObject>ReadFromVideoCamera.Program</StartupObject>
</PropertyGroup>
<ItemGroup>
    <Reference Include="Bytescout.BarCodeReader, Version=8.20.0.1340, Culture=neutral,
        <SpecificVersion>False</SpecificVersion>
        <HintPath>C:\Program Files\Bytescout BarCode Reader SDK\net2.00\Bytescout.BarCode
    </Reference>
    <Reference Include="System" />
    <Reference Include="System.Data" />
    <Reference Include="System.Deployment" />
    <Reference Include="System.Drawing" />
    <Reference Include="System.Windows.Forms" />
    <Reference Include="System.Xml" />
    <Reference Include="TouchlessLib, Version=1.0.0.0, Culture=neutral, processorArchit
        <SpecificVersion>False</SpecificVersion>
    </Reference>
</ItemGroup>
<ItemGroup>
    <Compile Include="MainForm.cs">
        <SubType>Form</SubType>
    </Compile>
    <Compile Include="MainForm.Designer.cs">
        <DependentUpon>MainForm.cs</DependentUpon>
    </Compile>
    <Compile Include="Program.cs" />
</ItemGroup>
<ItemGroup>
</ItemGroup>
<ItemGroup>
    <Content Include="TouchlessLib.dll">
    </Content>
    <Content Include="WebCamLib.dll">
        <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
</ItemGroup>
<ItemGroup>
    <EmbeddedResource Include="MainForm.resx">
        <DependentUpon>MainForm.cs</DependentUpon>
    </EmbeddedResource>
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
<!-- To modify your build process, add your task inside one of the targets below and
    Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>

```



```

<?xml version="1.0" encoding="utf-8"?>
<Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup>
    <Configuration Condition="'$(Configuration)' == ''">Debug</Configuration>
    <Platform Condition="'$(Platform)' == ''">x86</Platform>
    <ProductVersion>
    </ProductVersion>
    <SchemaVersion>2.0</SchemaVersion>
    <ProjectGuid>{9EACD999-022B-495A-BCDF-BB82C2534522}</ProjectGuid>
    <OutputType>WinExe</OutputType>
    <RootNamespace>ReadFromVideoCamera</RootNamespace>
    <AssemblyName>ReadFromVideoCamera</AssemblyName>
    <FileUpgradeFlags>
    </FileUpgradeFlags>
    <UpgradeBackupLocation />
    <PublishUrl>publish\</PublishUrl>
    <Install>true</Install>
    <InstallFrom>Disk</InstallFrom>
    <UpdateEnabled>>false</UpdateEnabled>
    <UpdateMode>Foreground</UpdateMode>
    <UpdateInterval>7</UpdateInterval>
    <UpdateIntervalUnits>Days</UpdateIntervalUnits>
    <UpdatePeriodically>>false</UpdatePeriodically>
    <UpdateRequired>>false</UpdateRequired>
    <MapFileExtensions>>true</MapFileExtensions>
    <ApplicationRevision>0</ApplicationRevision>
    <ApplicationVersion>1.0.0.%2a</ApplicationVersion>
    <IsWebBootstrapper>>false</IsWebBootstrapper>
    <UseApplicationTrust>>false</UseApplicationTrust>
    <BootstrapperEnabled>>true</BootstrapperEnabled>
    <OldToolsVersion>3.5</OldToolsVersion>
    <TargetFrameworkVersion>v4.0</TargetFrameworkVersion>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|x86'">
    <DebugSymbols>>true</DebugSymbols>
    <OutputPath>bin\x86\Debug\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <DebugType>full</DebugType>
    <PlatformTarget>x86</PlatformTarget>
    <ErrorReport>prompt</ErrorReport>
    <CodeAnalysisRuleSet>AllRules.ruleset</CodeAnalysisRuleSet>
    <CodeAnalysisIgnoreBuiltInRuleSets>>false</CodeAnalysisIgnoreBuiltInRuleSets>
    <CodeAnalysisIgnoreBuiltInRules>>false</CodeAnalysisIgnoreBuiltInRules>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Release|x86'">
    <OutputPath>bin\x86\Release\</OutputPath>
    <DefineConstants>TRACE</DefineConstants>
    <Optimize>true</Optimize>
    <DebugType>pdbonly</DebugType>
    <PlatformTarget>x86</PlatformTarget>
    <ErrorReport>prompt</ErrorReport>
    <CodeAnalysisRuleSet>AllRules.ruleset</CodeAnalysisRuleSet>
    <CodeAnalysisIgnoreBuiltInRuleSets>>false</CodeAnalysisIgnoreBuiltInRuleSets>
    <CodeAnalysisIgnoreBuiltInRules>>false</CodeAnalysisIgnoreBuiltInRules>
  </PropertyGroup>

```

```

<PropertyGroup>
  <StartupObject>ReadFromVideoCamera.Program</StartupObject>
</PropertyGroup>
<ItemGroup>
  <Reference Include="Bytescout.BarCodeReader, Version=8.20.0.1340, Culture=neutral,
    <SpecificVersion>False</SpecificVersion>
    <HintPath>C:\Program Files\Bytescout BarCode Reader SDK\net4.00\Bytescout.BarCode
  </Reference>
  <Reference Include="System" />
  <Reference Include="System.Data" />
  <Reference Include="System.Deployment" />
  <Reference Include="System.Drawing" />
  <Reference Include="System.Windows.Forms" />
  <Reference Include="System.Xml" />
  <Reference Include="TouchlessLib, Version=1.0.0.0, Culture=neutral, processorArchit
    <SpecificVersion>False</SpecificVersion>
  </Reference>
</ItemGroup>
<ItemGroup>
  <Compile Include="MainForm.cs">
    <SubType>Form</SubType>
  </Compile>
  <Compile Include="MainForm.Designer.cs">
    <DependentUpon>MainForm.cs</DependentUpon>
  </Compile>
  <Compile Include="Program.cs" />
</ItemGroup>
<ItemGroup>
</ItemGroup>
<ItemGroup>
  <Content Include="TouchlessLib.dll">
  </Content>
  <Content Include="WebCamLib.dll">
    <CopyToOutputDirectory>Always</CopyToOutputDirectory>
  </Content>
</ItemGroup>
<ItemGroup>
  <EmbeddedResource Include="MainForm.resx">
    <DependentUpon>MainForm.cs</DependentUpon>
  </EmbeddedResource>
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
<!-- To modify your build process, add your task inside one of the targets below and
  Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>

```

```
<?xml version="1.0" encoding="utf-8"?>
<Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003"
  <PropertyGroup>
    <Configuration Condition="'$(Configuration)' == 'Debug'>Debug</Configuration>
    <Platform Condition="'$(Platform)' == 'x86'>x86</Platform>
    <SchemaVersion>2.0</SchemaVersion>
    <ProjectGuid>{9EACD999-022B-495A-BCDF-BB82C2534522}</ProjectGuid>
    <OutputType>WinExe</OutputType>
    <RootNamespace>ReadFromVideoCamera</RootNamespace>
    <AssemblyName>ReadFromVideoCamera</AssemblyName>
    <FileUpgradeFlags>
    </FileUpgradeFlags>
    <UpgradeBackupLocation />
    <PublishUrl>publish\</PublishUrl>
    <Install>true</Install>
    <InstallFrom>Disk</InstallFrom>
    <UpdateEnabled>>false</UpdateEnabled>
    <UpdateMode>Foreground</UpdateMode>
    <UpdateInterval>7</UpdateInterval>
    <UpdateIntervalUnits>Days</UpdateIntervalUnits>
    <UpdatePeriodically>>false</UpdatePeriodically>
    <UpdateRequired>>false</UpdateRequired>
    <MapFileExtensions>>true</MapFileExtensions>
    <ApplicationRevision>0</ApplicationRevision>
    <ApplicationVersion>1.0.0.%2a</ApplicationVersion>
    <IsWebBootstrapper>>false</IsWebBootstrapper>
    <UseApplicationTrust>>false</UseApplicationTrust>
    <BootstrapperEnabled>>true</BootstrapperEnabled>
    <OldToolsVersion>3.5</OldToolsVersion>
    <TargetFrameworkVersion>v4.0</TargetFrameworkVersion>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|x86'>
    <DebugSymbols>>true</DebugSymbols>
    <OutputPath>bin\x86\Debug\</OutputPath>
    <DefineConstants>DEBUG;TRACE</DefineConstants>
    <DebugType>full</DebugType>
    <PlatformTarget>x86</PlatformTarget>
    <ErrorReport>prompt</ErrorReport>
    <CodeAnalysisRuleSet>AllRules.ruleset</CodeAnalysisRuleSet>
    <CodeAnalysisIgnoreBuiltInRuleSets>>false</CodeAnalysisIgnoreBuiltInRuleSets>
    <CodeAnalysisIgnoreBuiltInRules>>false</CodeAnalysisIgnoreBuiltInRules>
  </PropertyGroup>
  <PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Release|x86'>
    <OutputPath>bin\x86\Release\</OutputPath>
    <DefineConstants>TRACE</DefineConstants>
    <Optimize>>true</Optimize>
    <DebugType>pdbonly</DebugType>
    <PlatformTarget>x86</PlatformTarget>
    <ErrorReport>prompt</ErrorReport>
    <CodeAnalysisRuleSet>AllRules.ruleset</CodeAnalysisRuleSet>
    <CodeAnalysisIgnoreBuiltInRuleSets>>false</CodeAnalysisIgnoreBuiltInRuleSets>
    <CodeAnalysisIgnoreBuiltInRules>>false</CodeAnalysisIgnoreBuiltInRules>
  </PropertyGroup>
  <PropertyGroup>
    <StartupObject>ReadFromVideoCamera.Program</StartupObject>
  </PropertyGroup>
  <ItemGroup>
    <Reference Include="Bytescout.BarCodeReader, Version=8.20.0.1340, Culture=neutral,
      <SpecificVersion>False</SpecificVersion>
```

```

    <HintPath>C:\Program Files\Bytescout BarCode Reader SDK\net4.00\Bytescout.BarCode
</Reference>
<Reference Include="System" />
<Reference Include="System.Data" />
<Reference Include="System.Deployment" />
<Reference Include="System.Drawing" />
<Reference Include="System.Windows.Forms" />
<Reference Include="System.Xml" />
<Reference Include="TouchlessLib, Version=1.0.0.0, Culture=neutral, processorArchit
    <SpecificVersion>False</SpecificVersion>
</Reference>
</ItemGroup>
<ItemGroup>
    <Compile Include="MainForm.cs">
        <SubType>Form</SubType>
    </Compile>
    <Compile Include="MainForm.Designer.cs">
        <DependentUpon>MainForm.cs</DependentUpon>
    </Compile>
    <Compile Include="Program.cs" />
</ItemGroup>
<ItemGroup>
</ItemGroup>
<ItemGroup>
    <Content Include="TouchlessLib.dll">
    </Content>
    <Content Include="WebCamLib.dll">
        <CopyToOutputDirectory>Always</CopyToOutputDirectory>
    </Content>
</ItemGroup>
<ItemGroup>
    <EmbeddedResource Include="MainForm.resx">
        <DependentUpon>MainForm.cs</DependentUpon>
    </EmbeddedResource>
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
<!-- To modify your build process, add your task inside one of the targets below and
    Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>

```

ReadFromVideoCamera.VS2013.csproj

```

<?xml version="1.0" encoding="utf-8"?>
<Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
    <PropertyGroup>
        <Configuration Condition="'$(Configuration)' == ''">Debug</Configuration>
    </PropertyGroup>

```

```

<Platform Condition=" '$(Platform)' == 'x86' ">x86</Platform>
<SchemaVersion>2.0</SchemaVersion>
<ProjectGuid>{9EACD999-022B-495A-BCDF-BB82C2534522}</ProjectGuid>
<OutputType>WinExe</OutputType>
<RootNamespace>ReadFromVideoCamera</RootNamespace>
<AssemblyName>ReadFromVideoCamera</AssemblyName>
<FileUpgradeFlags>
</FileUpgradeFlags>
<UpgradeBackupLocation />
<PublishUrl>publish\</PublishUrl>
<Install>true</Install>
<InstallFrom>Disk</InstallFrom>
<UpdateEnabled>>false</UpdateEnabled>
<UpdateMode>Foreground</UpdateMode>
<UpdateInterval>7</UpdateInterval>
<UpdateIntervalUnits>Days</UpdateIntervalUnits>
<UpdatePeriodically>>false</UpdatePeriodically>
<UpdateRequired>>false</UpdateRequired>
<MapFileExtensions>>true</MapFileExtensions>
<ApplicationRevision>0</ApplicationRevision>
<ApplicationVersion>1.0.0.%2a</ApplicationVersion>
<IsWebBootstrapper>>false</IsWebBootstrapper>
<UseApplicationTrust>>false</UseApplicationTrust>
<BootstrapperEnabled>>true</BootstrapperEnabled>
<OldToolsVersion>3.5</OldToolsVersion>
<TargetFrameworkVersion>v4.0</TargetFrameworkVersion>
</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|x86' ">
<DebugSymbols>>true</DebugSymbols>
<OutputPath>bin\x86\Debug\</OutputPath>
<DefineConstants>DEBUG;TRACE</DefineConstants>
<DebugType>full</DebugType>
<PlatformTarget>x86</PlatformTarget>
<ErrorReport>prompt</ErrorReport>
<CodeAnalysisRuleSet>AllRules.ruleset</CodeAnalysisRuleSet>
<CodeAnalysisIgnoreBuiltInRuleSets>>false</CodeAnalysisIgnoreBuiltInRuleSets>
<CodeAnalysisIgnoreBuiltInRules>>false</CodeAnalysisIgnoreBuiltInRules>
</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|x86' ">
<OutputPath>bin\x86\Release\</OutputPath>
<DefineConstants>TRACE</DefineConstants>
<Optimize>>true</Optimize>
<DebugType>pdbonly</DebugType>
<PlatformTarget>x86</PlatformTarget>
<ErrorReport>prompt</ErrorReport>
<CodeAnalysisRuleSet>AllRules.ruleset</CodeAnalysisRuleSet>
<CodeAnalysisIgnoreBuiltInRuleSets>>false</CodeAnalysisIgnoreBuiltInRuleSets>
<CodeAnalysisIgnoreBuiltInRules>>false</CodeAnalysisIgnoreBuiltInRules>
</PropertyGroup>
<PropertyGroup>
<StartupObject>ReadFromVideoCamera.Program</StartupObject>
</PropertyGroup>
<ItemGroup>
<Reference Include="Bytescout.BarCodeReader, Version=8.20.0.1340, Culture=neutral,
<SpecificVersion>False</SpecificVersion>
<HintPath>C:\Program Files\Bytescout BarCode Reader SDK\net4.00\Bytescout.BarCode
</Reference>
<Reference Include="System" />
<Reference Include="System.Data" />
<Reference Include="System.Deployment" />

```

```

<Reference Include="System.Drawing" />
<Reference Include="System.Windows.Forms" />
<Reference Include="System.Xml" />
<Reference Include="TouchlessLib, Version=1.0.0.0, Culture=neutral, processorArchit
  <SpecificVersion>False</SpecificVersion>
</Reference>
</ItemGroup>
<ItemGroup>
  <Compile Include="MainForm.cs">
    <SubType>Form</SubType>
  </Compile>
  <Compile Include="MainForm.Designer.cs">
    <DependentUpon>MainForm.cs</DependentUpon>
  </Compile>
  <Compile Include="Program.cs" />
</ItemGroup>
<ItemGroup>
</ItemGroup>
<ItemGroup>
  <Content Include="TouchlessLib.dll">
  </Content>
  <Content Include="WebCamLib.dll">
    <CopyToOutputDirectory>Always</CopyToOutputDirectory>
  </Content>
</ItemGroup>
<ItemGroup>
  <EmbeddedResource Include="MainForm.resx">
    <DependentUpon>MainForm.cs</DependentUpon>
  </EmbeddedResource>
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
<!-- To modify your build process, add your task inside one of the targets below and
  Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>

```

ReadFromVideoCamera.VS2015.csproj

```

<?xml version="1.0" encoding="utf-8"?>
<Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003"
  <PropertyGroup>
    <Configuration Condition="'$(Configuration)' == ''">Debug</Configuration>
    <Platform Condition="'$(Platform)' == ''">x86</Platform>
    <SchemaVersion>2.0</SchemaVersion>
    <ProjectGuid>{9EACD999-022B-495A-BCDF-BB82C2534522}</ProjectGuid>
    <OutputType>WinExe</OutputType>
    <RootNamespace>ReadFromVideoCamera</RootNamespace>

```

```

<AssemblyName>ReadFromVideoCamera</AssemblyName>
<FileUpgradeFlags>
</FileUpgradeFlags>
<UpgradeBackupLocation />
<PublishUrl>publish\</PublishUrl>
<Install>true</Install>
<InstallFrom>Disk</InstallFrom>
<UpdateEnabled>>false</UpdateEnabled>
<UpdateMode>Foreground</UpdateMode>
<UpdateInterval>7</UpdateInterval>
<UpdateIntervalUnits>Days</UpdateIntervalUnits>
<UpdatePeriodically>>false</UpdatePeriodically>
<UpdateRequired>>false</UpdateRequired>
<MapFileExtensions>>true</MapFileExtensions>
<ApplicationRevision>0</ApplicationRevision>
<ApplicationVersion>1.0.0.%2a</ApplicationVersion>
<IsWebBootstrapper>>false</IsWebBootstrapper>
<UseApplicationTrust>>false</UseApplicationTrust>
<BootstrapperEnabled>>true</BootstrapperEnabled>
<OldToolsVersion>3.5</OldToolsVersion>
<TargetFrameworkVersion>v4.0</TargetFrameworkVersion>
</PropertyGroup>
<PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Debug|x86'">
  <DebugSymbols>>true</DebugSymbols>
  <OutputPath>bin\x86\Debug\</OutputPath>
  <DefineConstants>DEBUG;TRACE</DefineConstants>
  <DebugType>full</DebugType>
  <PlatformTarget>x86</PlatformTarget>
  <ErrorReport>prompt</ErrorReport>
  <CodeAnalysisRuleSet>AllRules.ruleset</CodeAnalysisRuleSet>
  <CodeAnalysisIgnoreBuiltInRuleSets>>false</CodeAnalysisIgnoreBuiltInRuleSets>
  <CodeAnalysisIgnoreBuiltInRules>>false</CodeAnalysisIgnoreBuiltInRules>
</PropertyGroup>
<PropertyGroup Condition="'$(Configuration)|$(Platform)' == 'Release|x86'">
  <OutputPath>bin\x86\Release\</OutputPath>
  <DefineConstants>TRACE</DefineConstants>
  <Optimize>>true</Optimize>
  <DebugType>pdbonly</DebugType>
  <PlatformTarget>x86</PlatformTarget>
  <ErrorReport>prompt</ErrorReport>
  <CodeAnalysisRuleSet>AllRules.ruleset</CodeAnalysisRuleSet>
  <CodeAnalysisIgnoreBuiltInRuleSets>>false</CodeAnalysisIgnoreBuiltInRuleSets>
  <CodeAnalysisIgnoreBuiltInRules>>false</CodeAnalysisIgnoreBuiltInRules>
</PropertyGroup>
<PropertyGroup>
  <StartupObject>ReadFromVideoCamera.Program</StartupObject>
</PropertyGroup>
<ItemGroup>
  <Reference Include="Bytescout.BarCodeReader, Version=8.20.0.1340, Culture=neutral,
    <SpecificVersion>False</SpecificVersion>
    <HintPath>C:\Program Files\Bytescout BarCode Reader SDK\net4.00\Bytescout.BarCode
  </Reference>
  <Reference Include="System" />
  <Reference Include="System.Data" />
  <Reference Include="System.Deployment" />
  <Reference Include="System.Drawing" />
  <Reference Include="System.Windows.Forms" />
  <Reference Include="System.Xml" />
  <Reference Include="TouchlessLib, Version=1.0.0.0, Culture=neutral, processorArchit
    <SpecificVersion>False</SpecificVersion>

```

```
</Reference>
</ItemGroup>
<ItemGroup>
  <Compile Include="MainForm.cs">
    <SubType>Form</SubType>
  </Compile>
  <Compile Include="MainForm.Designer.cs">
    <DependentUpon>MainForm.cs</DependentUpon>
  </Compile>
  <Compile Include="Program.cs" />
</ItemGroup>
<ItemGroup>
</ItemGroup>
<ItemGroup>
  <Content Include="TouchlessLib.dll">
  </Content>
  <Content Include="WebCamLib.dll">
    <CopyToOutputDirectory>Always</CopyToOutputDirectory>
  </Content>
</ItemGroup>
<ItemGroup>
  <EmbeddedResource Include="MainForm.resx">
    <DependentUpon>MainForm.cs</DependentUpon>
  </EmbeddedResource>
</ItemGroup>
<Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
<!-- To modify your build process, add your task inside one of the targets below and
    Other similar extension points exist, see Microsoft.Common.targets.
<Target Name="BeforeBuild">
</Target>
<Target Name="AfterBuild">
</Target>
-->
</Project>
```

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout BarCode Reader SDK](#)

[Explore documentation](#)

[Visit www.ByteScout.com](#)

or

[Get Your Free API Key for www.PDF.co Web API](#)

