



www.openwire.org www.mitov.com

Copyright Boian Mitov 2004 - 2014

Index

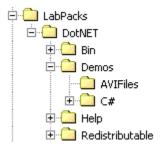
Installation	3
Where is VisionLab?	3
Creating a new VisionLab project in Visual C#	3
Installing the components on the Toolbox	
Adding the necessary assembly references to your application	
Creating a simple contour detection application	8
Distributing your application	14
Deploying your 32 bit application with the IPP DLLs	
Deploying your 64 bit application	

Installation

VideoLab comes with an installation program. Just start the installation by doubleclicking on the Setup.exe file and follow the installation instructions.

Where is VisionLab?

After the installation VideoLab is located under a single root directory. The default location is C:\Program Files\LabPacks or C:\Program Files (x86)\LabPacks on 64 bit systems. During the installation the user has the option to select alternative directory. Here is how the directory structure should look like after the installation:



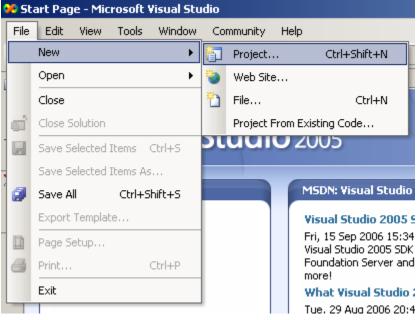
Under the "Demos" directory are located the demo files. The help files and the documentation are located under the "Help" directory. The component . NET 2.0/3.5/4.0 assemblies and the redistributable DLL files are located under the "Bin" directory. It is a great idea to start by opening and compiling the demo files. The demo projects ware developed with Visual C# 2005.

Creating a new VisionLab project in Visual C#

All of the examples in this manual start with creating a C# Windows .NET based project. The following chapters will assume that you have created the project and will teach you how to add specific VisionLab functionality.

Start by creating a new project.

From the VC++ menu, select | File | New | Project... |



In the "New Project" dialog select | Visual C# | Windows Application | Optionally you can select a project path and project name:

New Project					<u> </u>
Project types:		<u>T</u> emplates:		0101 0101	0-0- 0-0- 0-0-
Visual Basic Visual C# Windows Smart Devi Database Starter Kits Visual J# Visual C++ Other Project 1	5	Windows Application Windows Control Library Console Application Empty Project	Glass Library web Control Library row Windows Service row Crystal Reports Application		
A project for creat	ing an application wit	h a Windows user interface			
<u>N</u> ame:	WindowsApplication	n1			
Location:	C:\Documents and Settings\Boian Mitov\My Documents\Visual Studio 2005\Projects				
Solution Na <u>m</u> e:	WindowsApplication	n1	Create directory for solution		
			ОК	Cance	

Click OK.

.1

Installing the components on the Toolbox

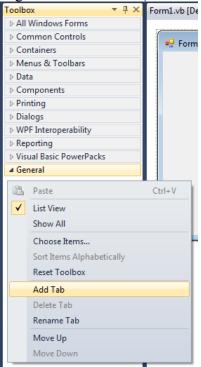
Before using the components in your project, you will have to install them on the component Toolbox.

The install in version 3.1 and up will automatically install the components on the toolbar, however if it fails, or if you have selected not to do so during the installation, here is a way to install the components manually:

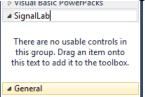
We assume that you have already created a project, and the toolbox with the .NET components has appeared.

Open the component toolbox and expand the General section.

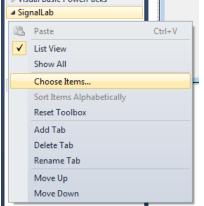
Right-click and select |Add Tab| from the menu:



Name the new tab "SignalLab":



Right-click on the SignalLab tab and select |Choose Items...| from the menu:



In the "Choose Toolbox Items" dialog select the components that belong to the Mitov.SignalLabBasic.DLL (You can order the components by "Assembly Name"). Press the "Spacebar" key to make sure all the components are selected:

Silverlight Componen	ts System	Workflow Components	System.Activiti	es Components
.NET Framework C	omponents	COM Components	WPF	Components
Name	Namespace	A	Assembly Name	Directory
✓ GenericFilter	Mitov.SignalLab	Ν	/litov.SignalLabBasic	. Global Asse
🗸 GenericGen	Mitov.SignalLab		/itov.SignalLabBasic	. Global Asse
🗸 GenericReal	Mitov.SignalLab		/itov.SignalLabBasic	. Global Asse
🗸 GenericRealGen	Mitov.SignalLab		/itov.SignalLabBasic	. Global Asse
🗸 GenericRealMatrix	Mitov.SignalLab		/itov.SignalLabBasic	. Global Asse
🗸 Pair	Mitov.SignalLab		/litov.SignalLabBasic	. Global Asse
BackgroundWorker	System.Compone	ntModel S	ystem	Global Asse
Component	System.Compone	ntModel S	ystem	Global Asse
🗸 EventLog	System.Diagnostic	is S	ystem	Global Asse
🛿 FileSystemWatcher	System.IO	S	ystem	Global Asse
MarchalPul/alusCom	System Compone	ntMadal C		Global Acco
jilter:				Clear
GenericFilter 🕵 Language: I	nvariant Language (Ir	nvariant Country)		<u>B</u> rowse

Click OK.

You should see the SignalLabBasic components on your toolbox:



Close and restart the Visual Studio IDE, then reopen the project.

On the "SignalLab" tab install Mitov.SignalLabAdditional.dll.

Repeat the same steps adding 4 more tabs named "MediaLab" "AudioLab", "VideoLab", and "VisionLab".

On the "MediaLab" tab install Mitov.MediaLabBasic.dll. On the "VideoLab" tab install Mitov.VideoLabBasic.dll and Mitov.VideoLabAdditional.dll. On the "VisionLab" tab install Mitov.VisionLab.dll.

Now you can start using the components in your .NET development.

Adding the necessary assembly references to your application

Visual studio will automatically add the assemblies being referenced when adding components to the project. If this mechanism fails, you can manually add the necessary assemblies as shown here:

In the "Solution Explorer" select the "References" node and right-click on it. From the menu select |Add Reference...|



Navigate to the Select the Mitov.VideoLabBasic.dll from the LabPacks\Bin\2.0 subdirectory and add the necessary assemblies.

Here is the list of necessary assemblies:

- For Mitov.BasicLab.DLL None.
- For Mitov.SignalLabBasic.DLL:
 - a. Mitov.BasicLab.DLL
- For Mitov.AudioLabBasic.DLL:
 - a. Mitov.BasicLab.DLL
- For Mitov.MediaLabBasic.DLL:

- a. Mitov.BasicLab.DLL
- b. Mitov.AudioLabBasic.DLL:
- For Mitov.VideoLabBasic.DLL:
 - a. Mitov.AudioLabBasic.DLL
 - b. Mitov.BasicLab.DLL
- For Mitov.VideoLabAdditional.DLL:
 - a. Mitov.VideoLabBasic.DLL
 - b. Mitov.AudioLabBasic.DLL
 - c. Mitov.BasicLab.DLL
- For Mitov.VisionLab.DLL:
 - a. Mitov.VideoLabAdditional.DLL
 - b. Mitov.VideoLabBasic.DLL
 - c. Mitov.AudioLabBasic.DLL
 - d. Mitov.BasicLab.DLL

Creating a simple contour detection application

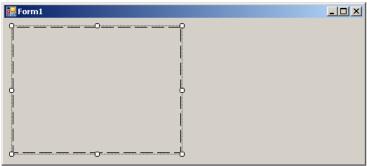
Create and setup a new project as described in the "Creating a new VideoLab project in Visual C#" chapter.

From the "VideoLab" tab on the Toolbox select and drop on the form the following two <u>components</u>:

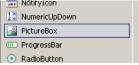


🚰 - FindContours

Arrange the ImageDisplay as shown here:



From the "Toolbox" switch to the "Common Controls" group and select the "PictureBox" component:



Place the component on the form and arrange it as shown here:

🖶 Form1	

Select the aviPlayer1 component on the form editor:

🕎 aviPlayer1	🤗 canny 1	😤 findContours1
--------------	-----------	-----------------

In the "Properties" palette go to the "OutputPin" property and click on the "..." button:

Pr	operties	+ ₽ ×
aviPlayer1 Mitov.VideoLab.AVIPlaye 🕶		
	PumpPriority	0
	RestartOnNewFi	True
Ξ	OpenWire	
	AudioOutputPin	(Disconnected)
	ClockPin	(Disconnected)
	EnablePin	OpenWire.Prox
	FileNamePin	(Disconnected)
	OutputPin	connected)
	PausePin	OpenWire.Prox
	ProgressPin	(Disconnected)

In the "OpenWire editor" check the following pins and click OK:

JUCPIN		
Component	Connected To	
imageDisplay1		
canny1		OpenWir
findContours1		
		Restore
		<mark>}</mark> €Link to all
		💥 Unlink all
		After Pin
		ОК
		Cancel
	imageDisplay1 canny1	Component Connected To imageDisplay1 canny1

In the "Properties" palette go to the "FileName" property and set a video file to play:

Pr	operties	- ₽	×		
a٩	aviPlayer1 Mitov.VideoLab.AVIPlaye 🕶				
•	2↓ 🗉 🖋				
	ClockSource	Internal			
	Enabled	True			
	FileName	6-indeo3.2.avi			
	InitialFrame	0			
	Loop	False			
	Paused	False			

Select the cannyl component on the form editor:

AVE) International		
🕎 aviPlayer1	Second Second	😤 findContours1
· · ·		· · · · · · · · · · · · · · · · · · ·

In the "Properties" palette go to the "OutputPin" property and click on the "..." button:

	Properties 🚽 🗸				
l	canny1 Mitov.VisionLab.Canny -				
	2↓ □				
		Modifiers	Private 🔺		
	Ξ	Misc			
		Enabled	True		
		HighThreshold	90		
		LowThreshold	20		
	Ŧ	WorkArea	Mitov.VideoLab		
1	Ξ	OpenWire			
		EnablePin	(Disconnected)		
		InputPin	aviPlayer1.Out		
		OutputPin	connected)		
			<u>_</u>		
	Π	utnutPin			

In the "OpenWire editor" check the following pin and click OK:

n the "OpenWire editor" check the following pin and click OK:				
🕸 Edit canny1.OutputPin				
	,			
Pin	Component	Connected To		
InputPin	imageDisplay1	aviPlayer1.OutputPin		
🗹 InputPin	findContours1		OpenWire	
			Restore	
			}{Link to all	
			💥 Unlink all	
			After Pin	
			Mancortin	
			ОК	
4			Cancel	
1				

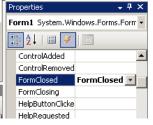
In the "Properties" palette go to the "FileName" property and set a video file to play:



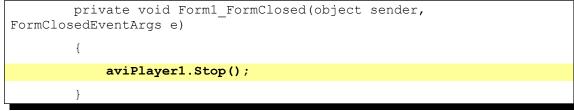
Select the form:

🔜 Form1	

In the "Properties" palette switch to events and double click on the FormClosed event to generate handler:



Add the highlighted line in the source file:



Select the findContours1 component on the form editor:

🐨 aviPlayer1 🞇 canny1 🖓 findContours1

In the "Properties" palette switch to events and double click on the Contours event to generate handler:



Add the highlighted line in the source file:

```
bool m ShowContours = true;
        private void findContours1 Contours(object Sender,
Mitov.VisionLab.ContoursEventArgs Args)
        {
            Mitov.VisionLab.Contours aContours = Args.Contours;
            int aContoursCount = aContours.Count;
            if (m ShowContours)
            {
                System.Drawing.Bitmap ABitmap = new
System.Drawing.Bitmap(240, 180);
                Graphics g = Graphics.FromImage(ABitmap);
                System.Drawing.SolidBrush ABrush = new
SolidBrush(Color.White);
                g.FillRectangle(ABrush, 0, 0, 240, 180);
                System.Drawing.Pen APen;
                System.Drawing.Pen AGgeenPen = new
System.Drawing.Pen(Color.Green, 1);
                System.Drawing.Pen ABluePen = new
System.Drawing.Pen(Color.Blue, 1);
                System.Drawing.Pen ARedPen = new
System.Drawing.Pen(Color.Red, 1);
                for (int i = 0; i < aContoursCount; i++)</pre>
                {
                    Mitov.VisionLab.Contour aContour =
aContours.get Items(i);
                    if (aContour.ContourType ==
Mitov.VisionLab.ContourType.Outer)
                        APen = AGgeenPen;
                    else
```

November 13, 2014

VisionLab .NET 7.5

APen = ABluePen;			
<pre>int ACount = aContour.Count;</pre>			
<pre>for (int j = 1; j < ACount; j++)</pre>			
<pre>g.DrawLine(APen, aContour.get_Items(j - 1), aContour.get_Items(j));</pre>			
g.DrawRectangle(ARedPen, aContours.get_Items(i).BoundRect);			
}			
<pre>pictureBox1.Image = ABitmap;</pre>			
<pre>m_ShowContours = false;</pre>			
}			
}			

From the "Toolbox" switch to the "Components" group and select the "Timer" component:

	1	
<i>,</i>	SerialPort	
	ServiceController	
Ö	Timer	
🗄 Printing		
a n	islage	

Select the timer1 component on the form editor:

🐺 aviPlayer1	🞇 canny1	🚏 findContours1	👸 timer 1
--------------	----------	-----------------	-----------

Enable the timer, and set the Interval to 1000:

Properties 🗸 🕂 🗙				
timer1 System.Windows.Forms.Time -				
2↓ 💷 🖋 । 🖻				
🗆 Behavior				
	Enabled	True 🔹		
	Interval	1000		
	Data			
Œ	(ApplicationSettii			

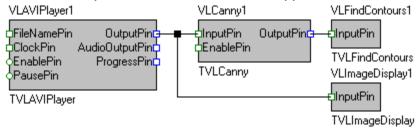
Double-click on the timer1 component, and add the highlighted lines in the source file:

	<pre>private void timer1_Tick(object sender, EventArgs e)</pre>
	{
	<pre>m_ShowContours = true;</pre>
_	}

Compile and run the application. You should see result similar to this one:



Here are the OpenWire connections in this application:



Distributing your application

Once you have finished the development of your application you most likely will need to distribute it to other systems. Version 5.0.2 and higher of the library will move all the necessary DLL files in the Release directory of your project. You will only need to distribute the files in the directory. To use this feature, make sure that the "Copy Local" property is set for all the assemblies from the library. Please check with the Visual Studio help for your version of Video Studio on how to configure assemblies as private assemblies.

Deploying your 32 bit application with the IPP DLLs

The compiled applications can be deployed to the target system by simply copying the executable. The application will work, however the performance can be improved by also copying the Intel IPP DLLs provided with the library.

The DLLs are under the [install path]\LabPacks\IppDLL\Win32 directory([install path] is the location where the library was installed).

In 32 bit Windows to deploy IPP, copy the files to the [Windows]\System32 directory on the target system.

In 64 bit Windows to deploy IPP, copy the files to the [Windows]\SysWOW64 directory on the target system.

[Windows] is the Windows directory - usually C:\WINNT or C:\WINDOWS This will improve the performance of your application on the target system.

Deploying your 64 bit application

The current version of the library requires when deploying 64 bit applications, the Intel IPP DLLs to be deployed as well.

November 13, 2014

The DLLs are under the [install path]\LabPacks\IppDLL\Win64 directory([install path] is the location where the library was installed).

To deploy IPP, copy the files to the [Windows]\System32 directory on the target system. [Windows] is the Windows directory - usually C:\WINNT or C:\WINDOWS This will improve the performance of your application on the target system.