

BasicVideo 4.5

.NET
Quick Start



www.openwire.org
www.mitov.com

Copyright Boian Mitov 2004 - 2010

Index

Installation.....	2
Where is VideoLab?.....	2
Creating a new VideoLab project in Visual C#	2
Installing the VideoLab components on the Toolbox.....	4
Adding the necessary assembly references to your application	7
Developing under 64 bit Windows.....	7
Creating a simple video player using DirectShow components	9
Distributing your application	12

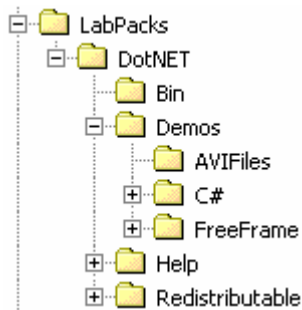
Installation

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

Where is VideoLab?

After the installation VideoLab is located under a single root directory. The default location is C:\Program Files\LabPacks. During the installation the user has the option to select alternative directory.

Here is how the directory structure should look like after the installation:

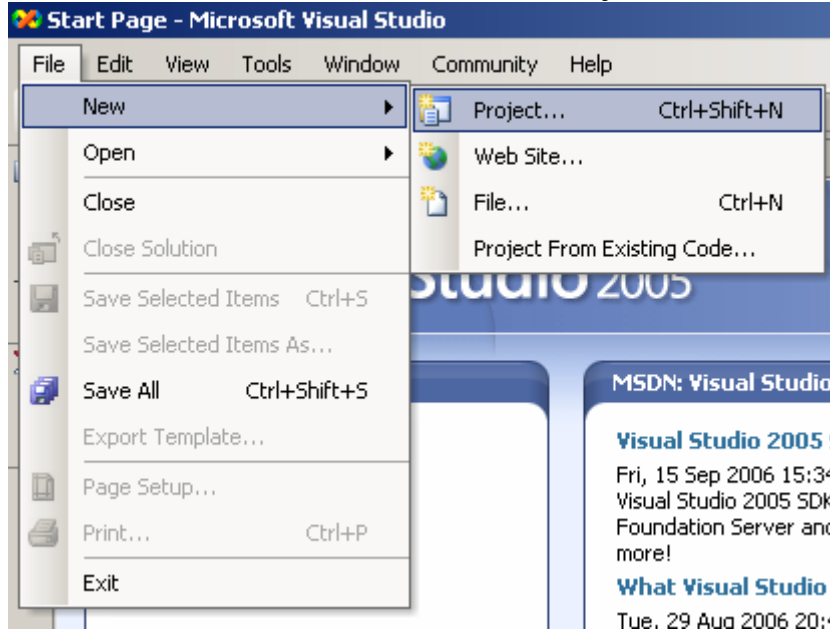


Under the “Demos” directory are located the demo files. The help files and the documentation are located under the “Help” directory. The component .NET 2.0/3.5 assemblies and the redistributable DLL/BPL files are located under the “Bin” directory. The “Redistributable” directory contains the installer for the Microsoft CRT libraries. They have to be present on any system where you plan to use VideoLab. It is a great idea to start by opening and compiling the demo files. The demo projects were developed with Visual C# 2005.

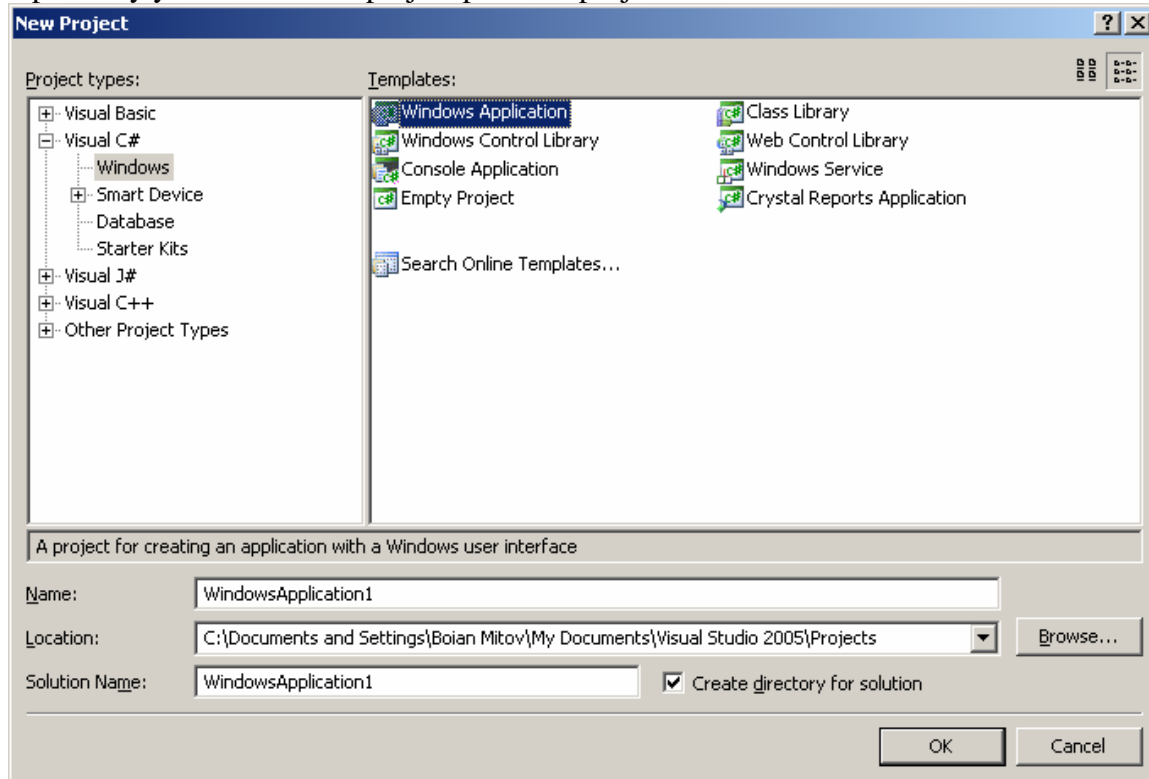
Creating a new VideoLab project in Visual C#

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 VideoLab 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:



Click OK.

Installing the VideoLab components on the Toolbox

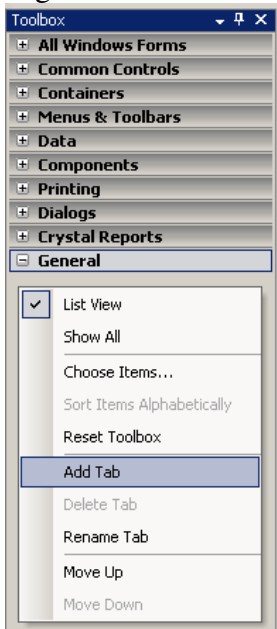
Before using the components in your project, you will have to install them on the component Toolbox.

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:

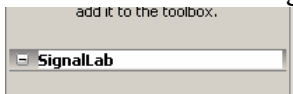
We assume that you have already created a project, and the toolbox with the .NET components has appeared.

Open the component toolbox and expand the General section.

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

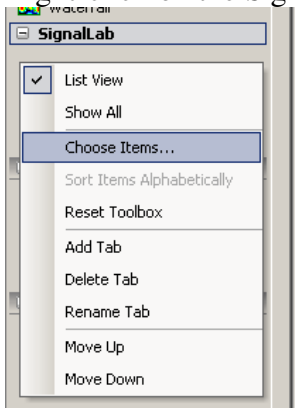


Name the new tab “SignalLab”:

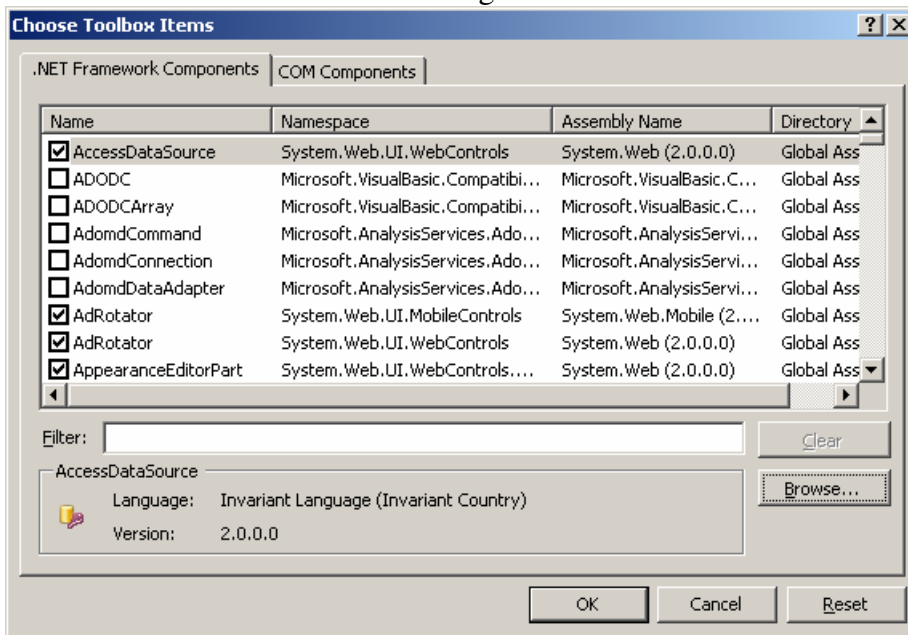


Repeat the same steps and add 3 more tabs named “AudioLab”, “MediaLab”, and “VideoLab”.

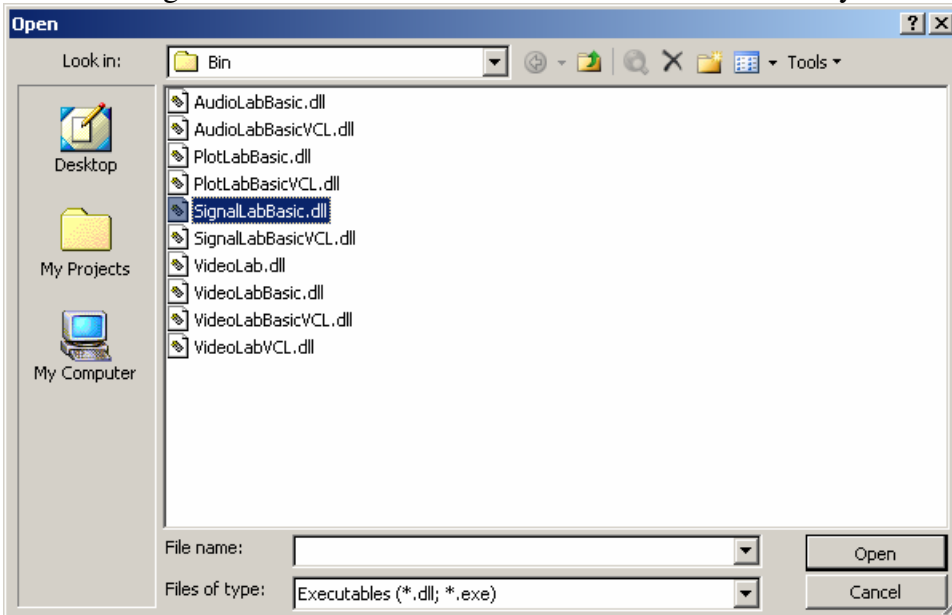
Right-click on the SignalLab tab and select |Choose Items...| from the menu:



In the “Choose Toolbox Items” dialog click on the “Browse...” button:

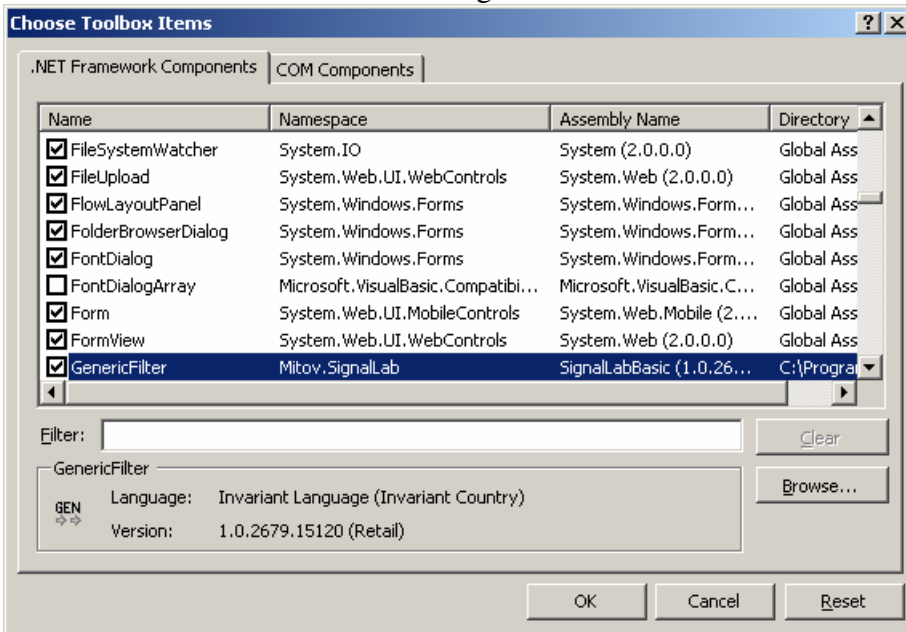


Select the SignalLabBasic.dll from the LabPacks\Bin subdirectory:

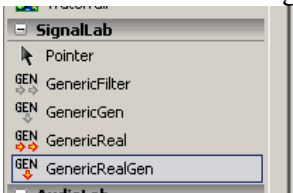


Click "Open".

In the "Choose Toolbox Items" dialog click OK.



You should see the SignalLab components on your toolbox:



Continue repeating the same steps and install the following assemblies:

On the “MediaLab” tab install MediaLabBasic.dll.

On the “AudioLab” tab install AudioLabBasic.dll.

On the “VideoLab” tab install VideoLabBasic.dll.

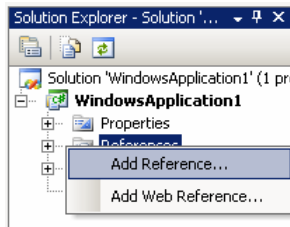
Now you can start using the components in your .NET development.

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 VideoLabBasic.dll from the LabPacks\Bin subdirectory and add the necessary assemblies.

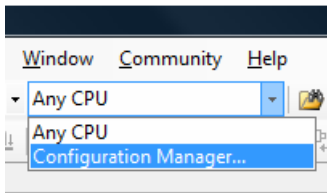
Here is the list of necessary assemblies:

- SignalLabBasic.DLL:
 - a. SignalLabBasicVCL.DLL
- MediaLabBasic.DLL:
 - a. MediaLabBasicVCL.DLL
 - b. SignalLabBasicVCL.DLL
 - c. SignalLabBasic.DLL
- AudioLabBasic.DLL:
 - a. AudioLabBasicVCL.DLL
 - b. SignalLabBasicVCL.DLL
 - c. SignalLabBasic.DLL
- VideoLabBasic.DLL:
 - a. VideoLabBasicVCL.DLL
 - b. AudioLabBasicVCL.DLL
 - c. AudioLabBasic.DLL
 - d. SignalLabBasicVCL.DLL
 - e. SignalLabBasic.DLL

Developing under 64 bit Windows

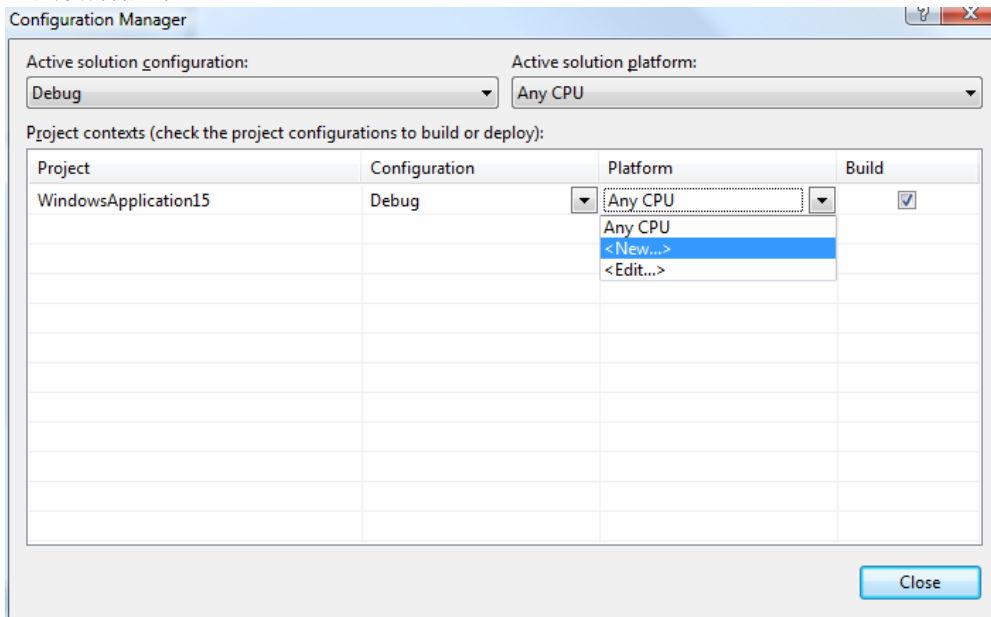
When developing projects under 64 bit Windows you will have to manually specify the Win32 target.

Click on the down arrow button of the target drop down:

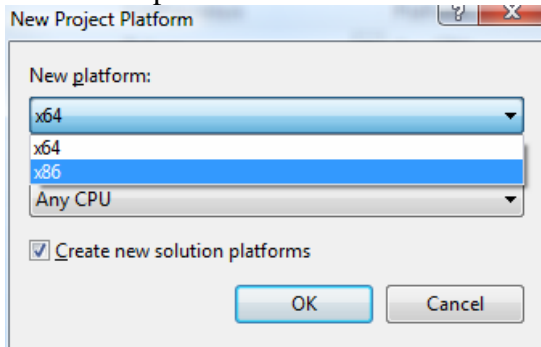


Select “Configuration Manager...”

In the “Configuration Manager” click in the down arrow of the platform and select “<New...>”:

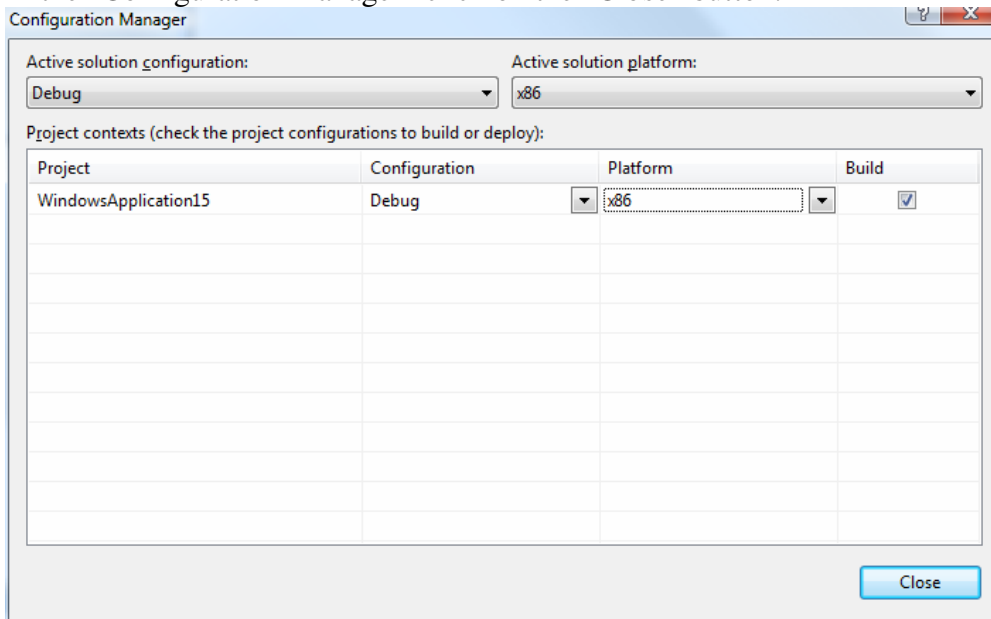


Select new platform x86:



Click Ok.

In the “Configuration Manager” click on the “Close” button:



Creating a simple video player using DirectShow components

WARNING: In order to run the application in this example you must have DirectX 9.0 or higher installed! This is very much true for most modern systems, however we will recommend checking to make sure your system is up to date.

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 components:

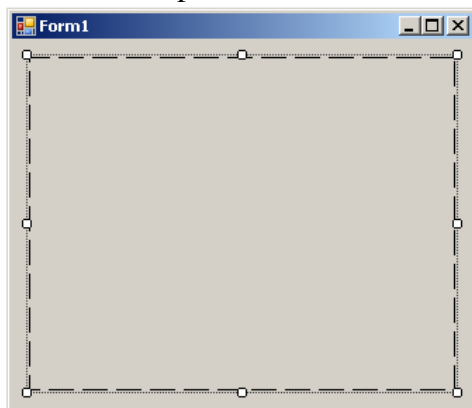


- DSVideoPlayer

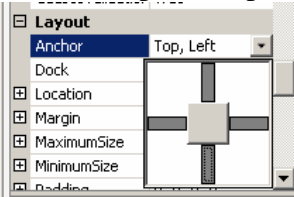


- DSImageDisplay

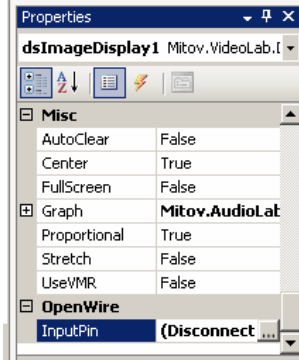
Place the component on the form and arrange it as shown here:



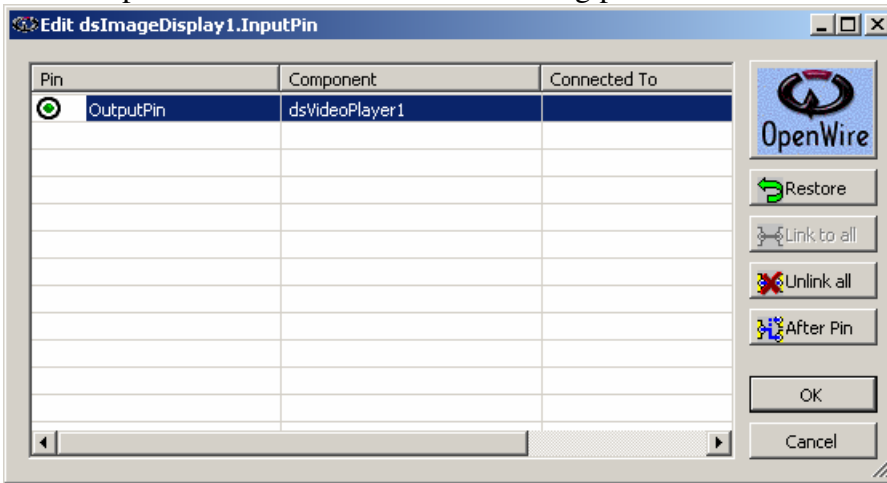
In the “Properties” palette set the Anchor property as shown here:



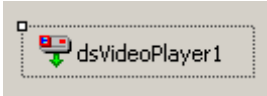
In the “Properties” palette go to the “InputPin” property and click on the “...” button:



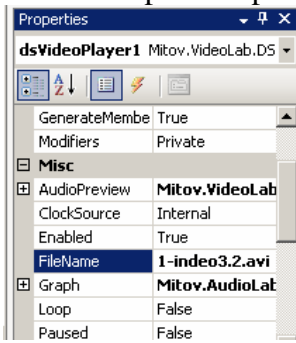
In the “OpenWire editor” check the following pin and click OK:



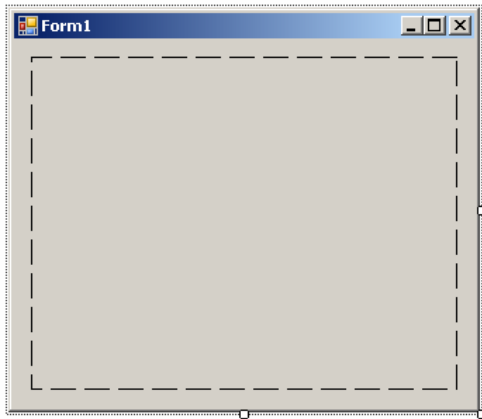
Select the dsVideoPlayer1 component on the form editor:



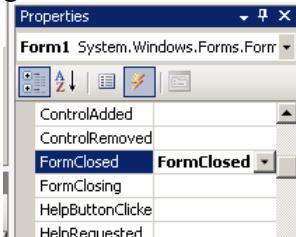
In the “Properties” palette go to the “FileName” property and set a video file to play:



Select the form:



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



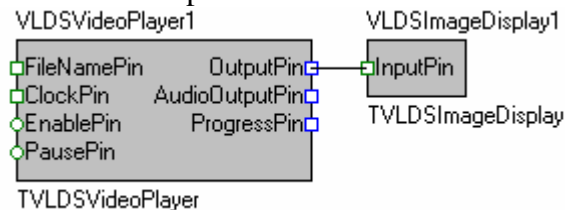
Add the highlighted line in the source file:

```
private void Form1_FormClosed(object sender,
FormClosedEventArgs e)
{
    dsVideoPlayer1.Stop();
}
```

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



Congratulations! You have just created your first VideoLab application. 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 3.1 and higher of the library will move all the necessary BPL and 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.

The executable also will rely on the Microsoft C++ RTL files. They are usually installed automatically when .NET 2.0/3.5 is installed, however in case you experience any problems you can download and install "Microsoft Visual C++ 2005 Redistributable Package (x86)" - vc_redist_x86.exe from Microsoft. This will ensure that you have the necessary RTL files.