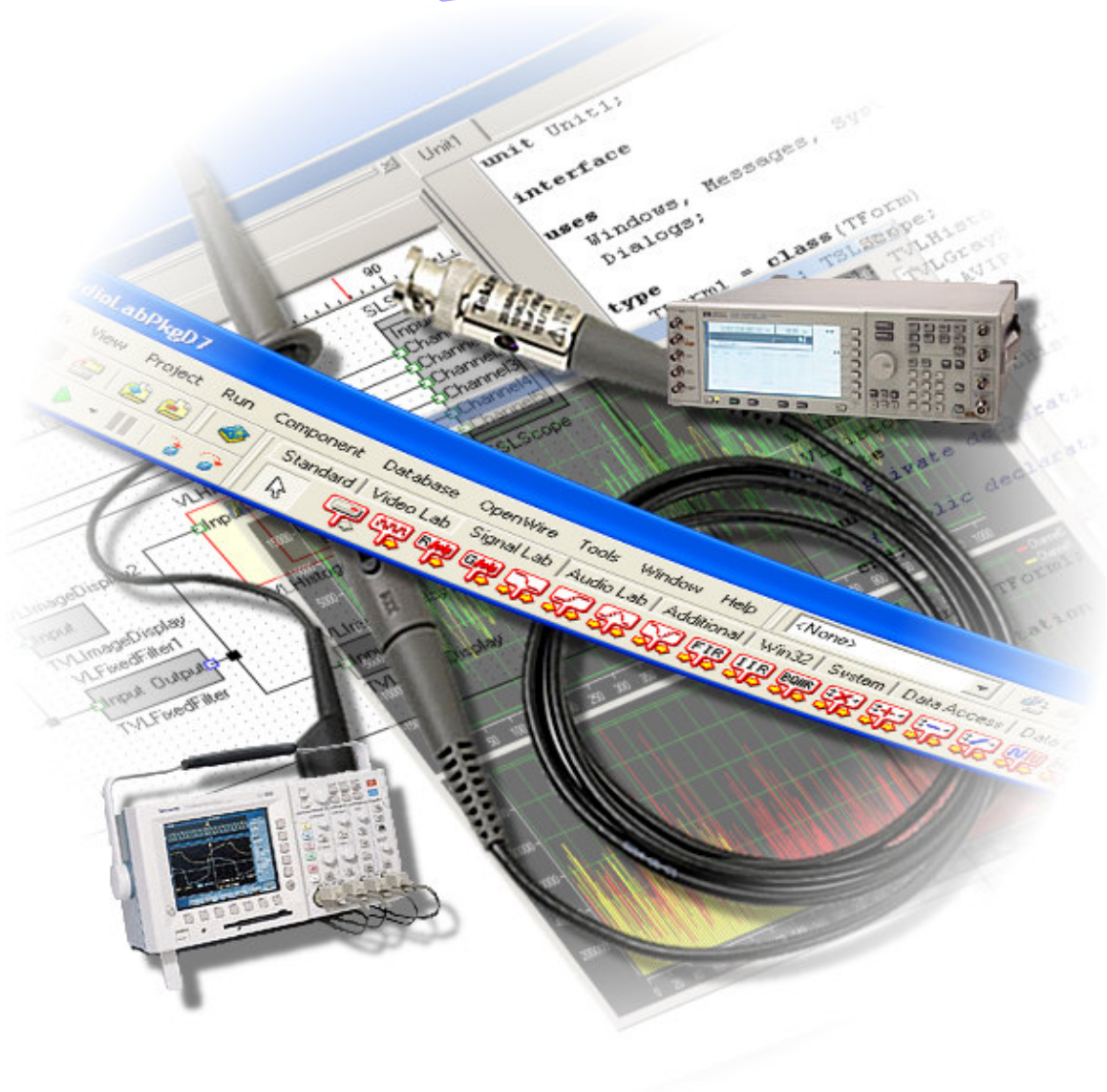


# SignalLab 4.5

**.NET**  
**Quick Start**



[www.openwire.org](http://www.openwire.org)  
[www.mitov.com](http://www.mitov.com)

Copyright Boian Mitov 2004 - 2010

# Index

Installation .....	2
Where is SignalLab? .....	2
Creating a new SignalLab project in Visual C# .....	2
Installing the SignalLab components on the Toolbox .....	4
Adding the necessary assembly references to your application .....	10
Developing under 64 bit Windows.....	11
Creating a simple video player using DirectShow components .....	12
Distributing your application .....	16

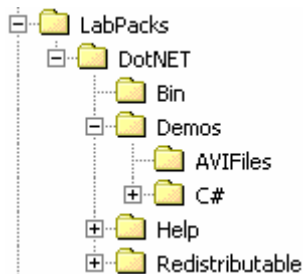
## Installation

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

## Where is SignalLab?

After the installation AudioLab 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:

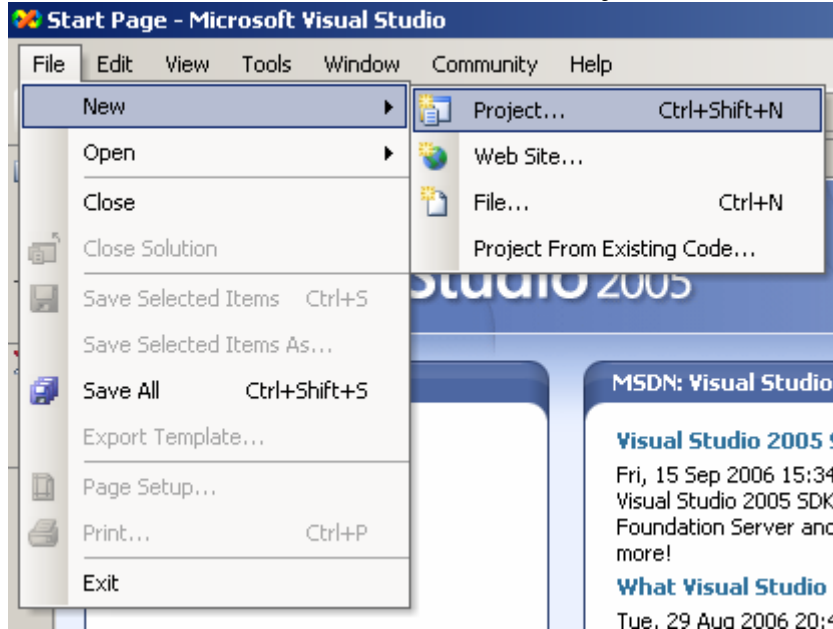


Under the “Demos” directory are located the demo files. The help files and the documentation are located under the “Help” directory. The component .NET 2.0/3.5 assemblies and the redistributable DLL/BPL files are located under the “Bin” directory. The “Redistributable” directory contains the installer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

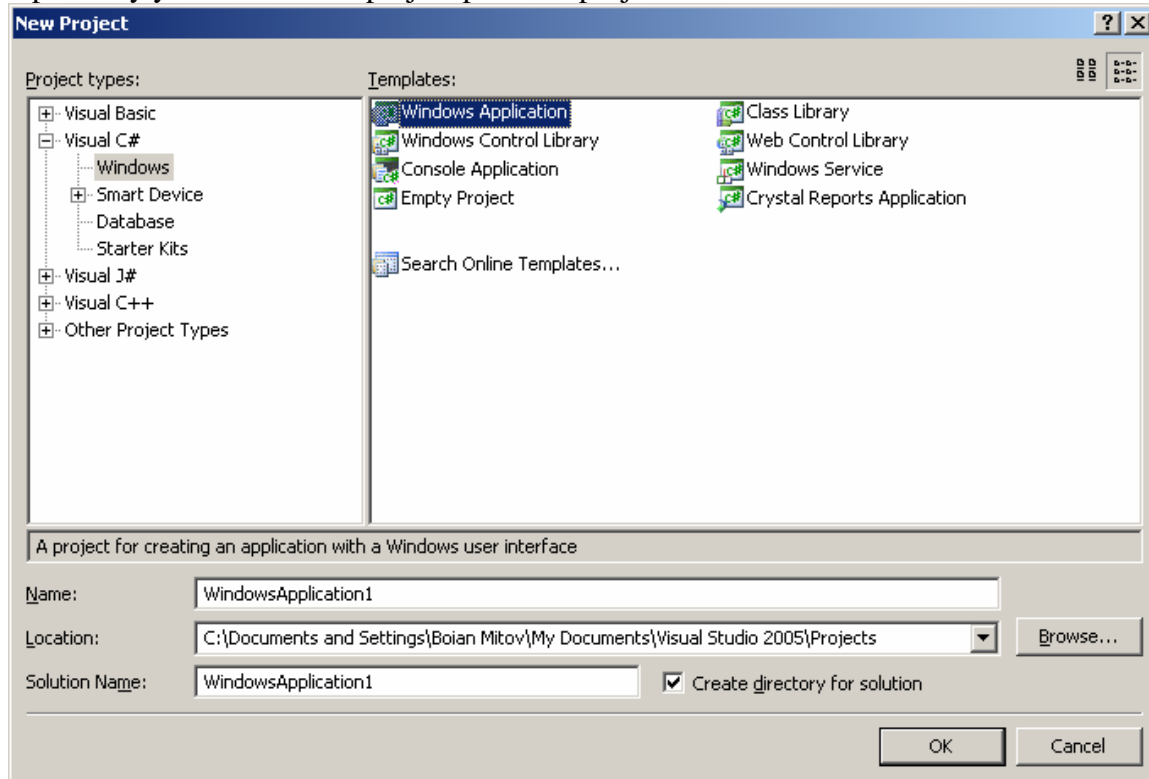
## Creating a new SignalLab project in Visual C#

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 SignalLab 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 SignalLab components on the Toolbox

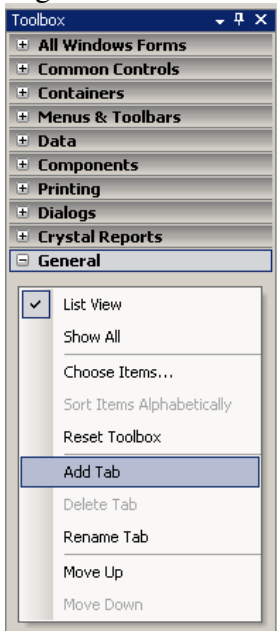
Before using the components in your project, you will have to install them on the component Toolbox.

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:

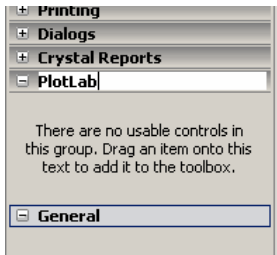
We assume that you have already created a project, and the toolbox with the .NET components has appeared.

Open the component toolbox and expand the General section.

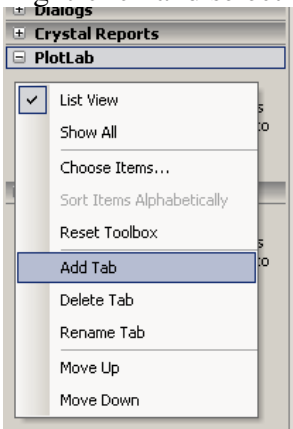
Right-click and select |Add Tab| from the menu:



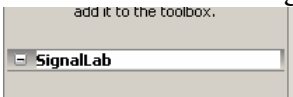
Name the new tab “PlotLab”:



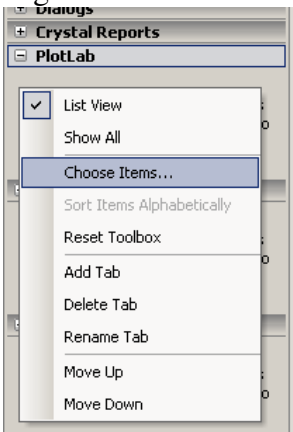
Right-click and select |Add Tab| from the menu:



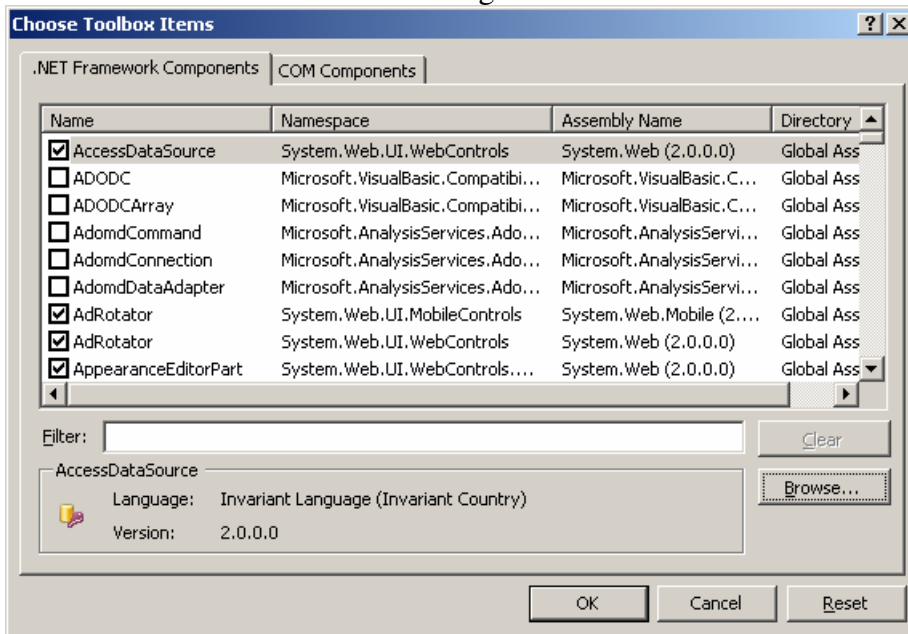
Name the new tab “SignalLab”:



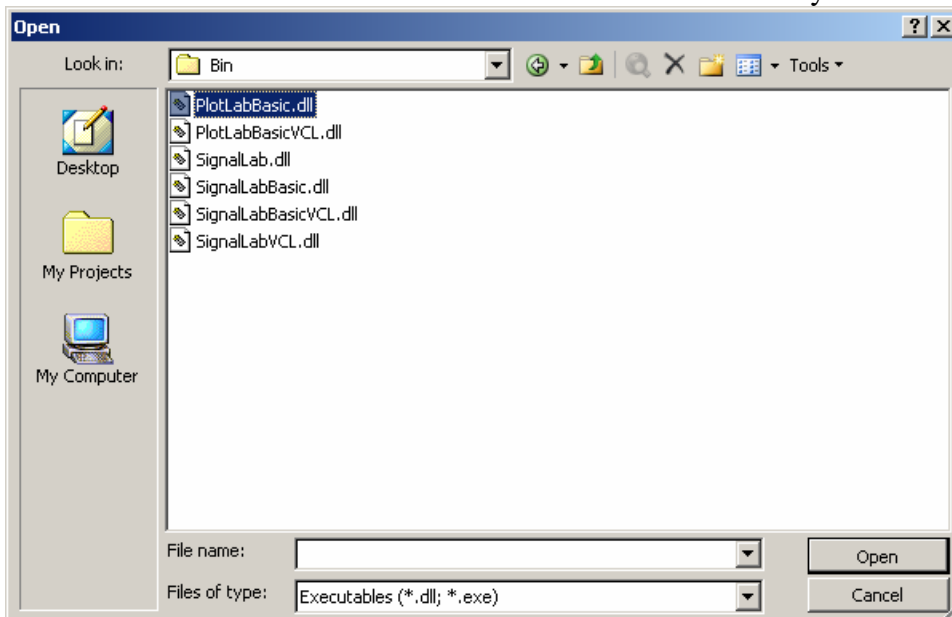
Right-click on the PlotLab tab and select |Choose Items...| from the menu:



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

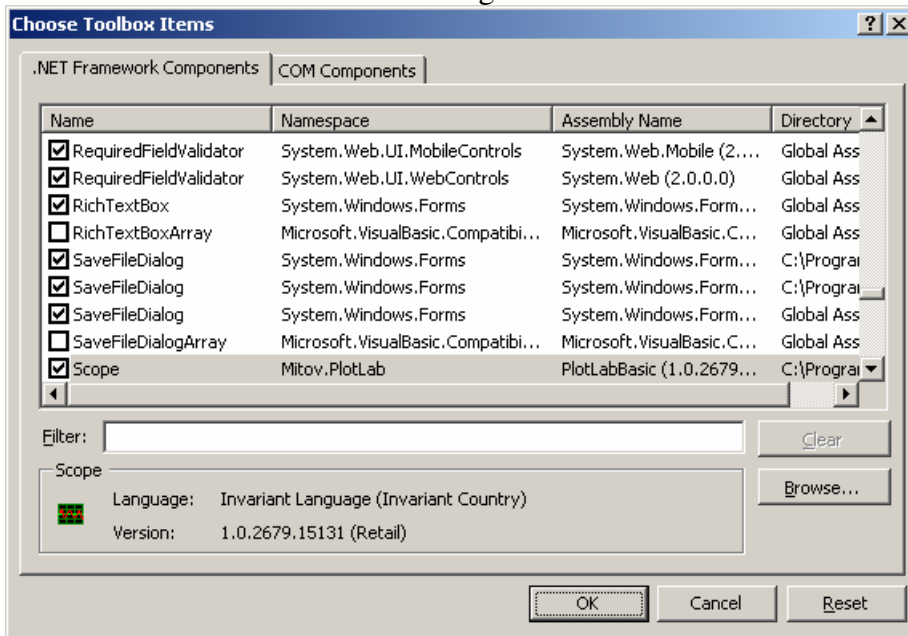


Select the PlotLabBasic.dll from the LabPacks\Bin subdirectory:

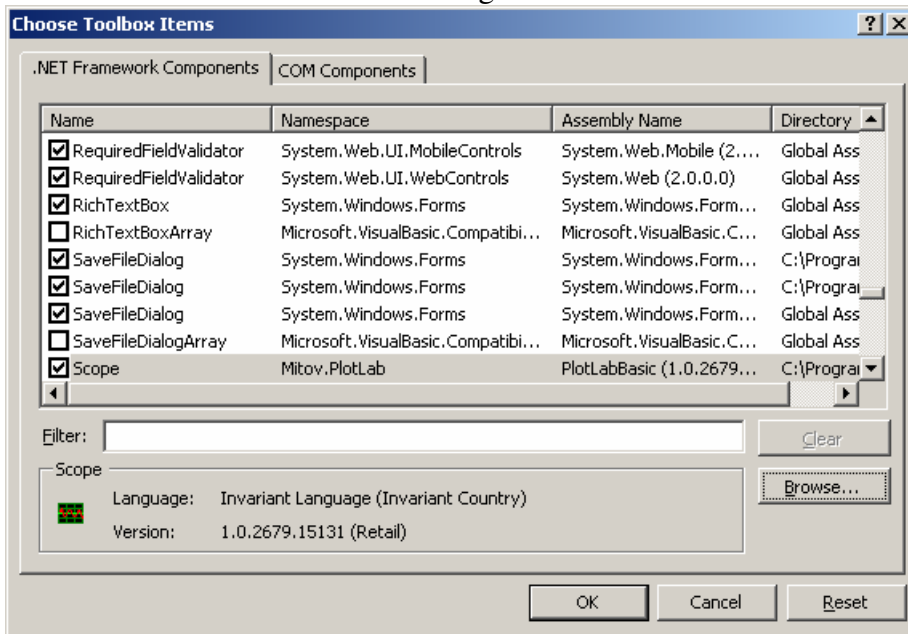


Click “Open”.

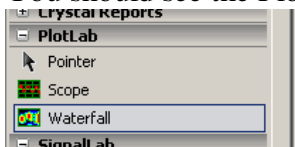
In the “Choose Toolbox Items” dialog click OK.



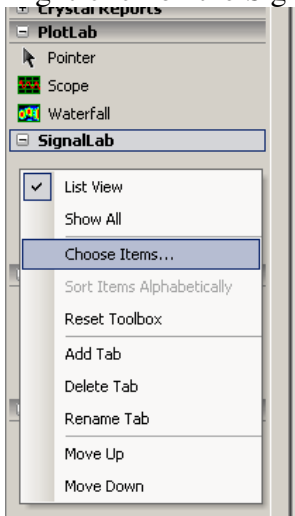
In the “Choose Toolbox Items” dialog click on the “Browse...” button:



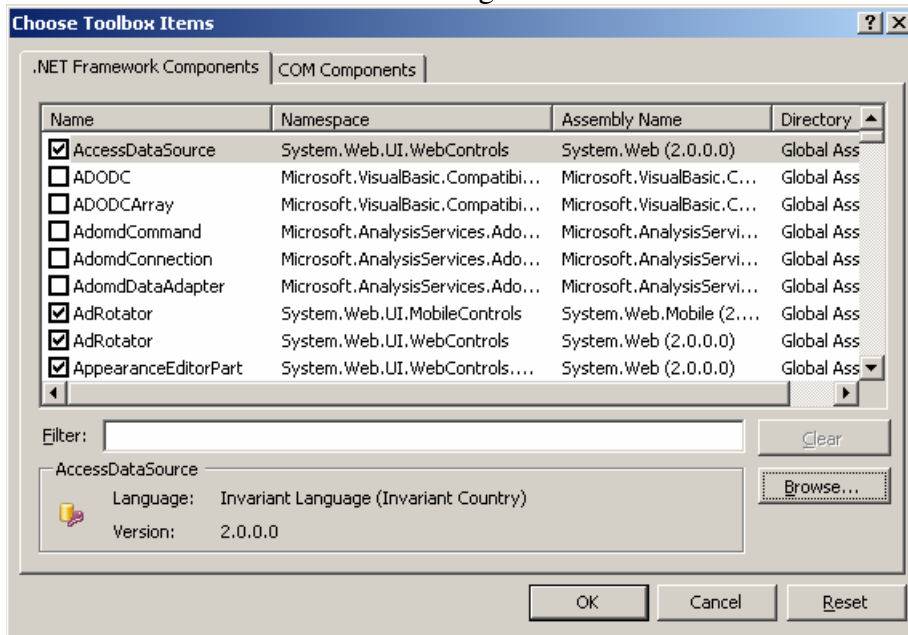
You should see the PlotLab components on your toolbox:



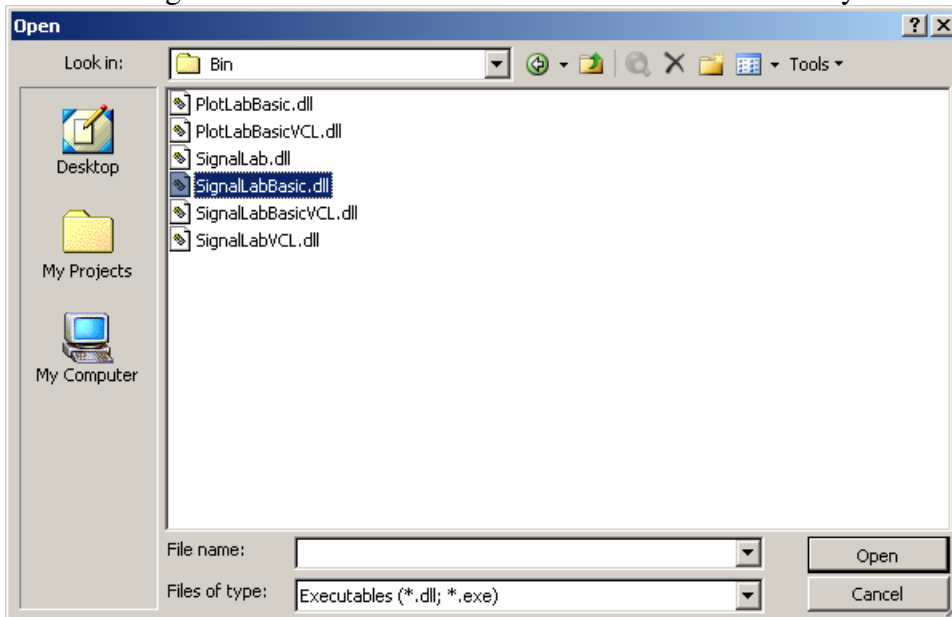
Right-click on the SignalLab tab and select |Choose Items...| from the menu:



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

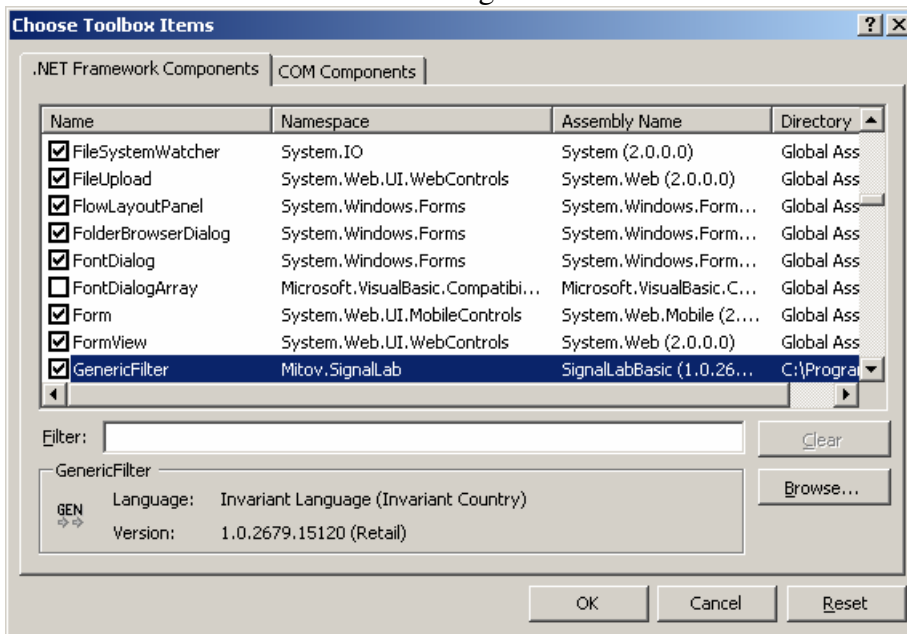


Select the SignalLabBasic.dll from the LabPacks\Bin subdirectory:

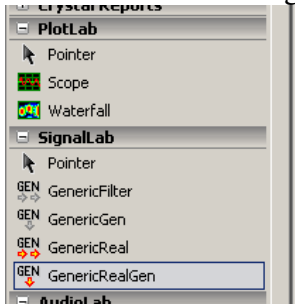


Click "Open".

In the "Choose Toolbox Items" dialog click OK.



You should see the SignalLab components on your toolbox:



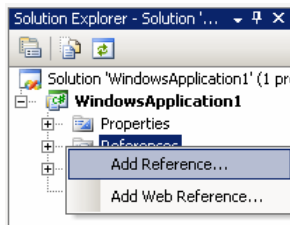
Continue repeating the same steps and install the following assemblies:  
On the “SignalLab” tab install SignalLabAdditional.dll and SignalLab.dll.  
Create a “TimingLab” tab and install TimingLabBasic.dll in it.

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

Here is the list of necessary assemblies:

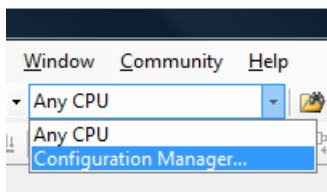
- For PlotLabBasic.DLL:
  - a. PlotLabBasicVCL.DLL
  - b. SignalLabBasicVCL.DLL
  - c. SignalLabBasic.DLL
- TimingLabBasic.DLL:
  - a. TimingLabBasicVCL.DLL
  - b. SignalLabBasicVCL.DLL
  - c. SignalLabBasic.DLL
- For SignalLabBasic.DLL:
  - a. SignalLabBasicVCL.DLL
- For SignalLabAdditional.DLL:
  - a. SignalLabAdditionalVCL.DLL

- b. SignalLabBasicVCL.DLL
  - c. SignalLabBasic.DLL
- For SignalLab.DLL:
  - a. SignalLabBasicVCL.DLL
  - b. SignalLabAdditional.DLL
  - c. SignalLabAdditionalVCL.DLL
  - d. SignalLabBasicVCL.DLL
  - e. SignalLabBasic.DLL

## Developing under 64 bit Windows

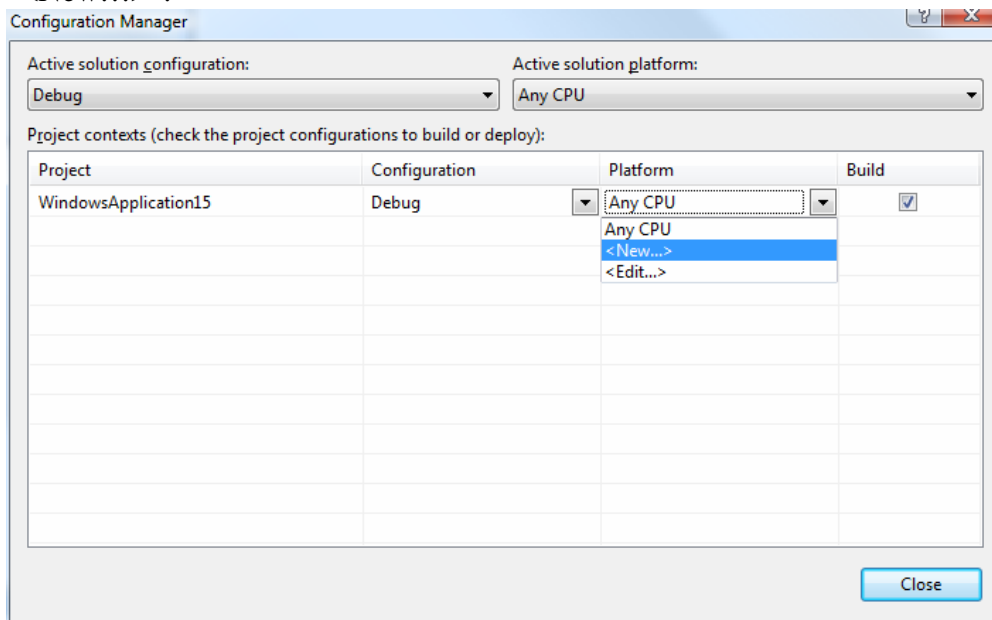
When developing projects under 64 bit Windows you will have to manually specify the Win32 target.

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

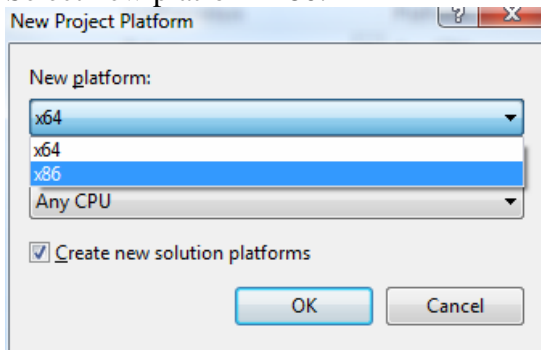


Select “Configuration Manager...”

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

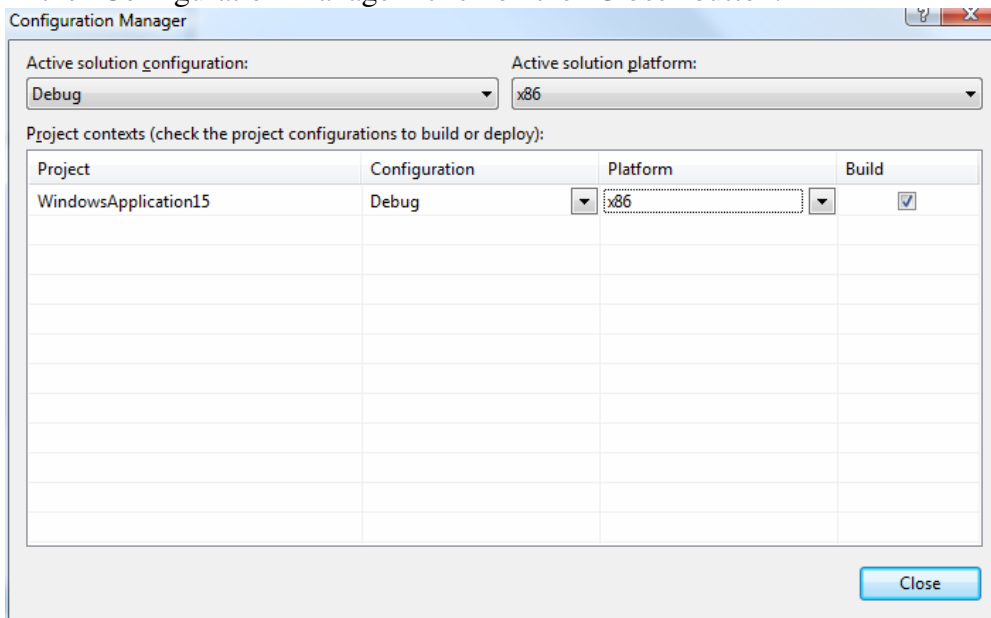


Select new platform x86:



Click Ok.

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



## Creating a simple video player using DirectShow components

Create and setup a new project as described in the “Creating a new SignalLab project in Visual C#” chapter.

From the “SignalLab” tab on the Toolbox select and drop on the form the following component:



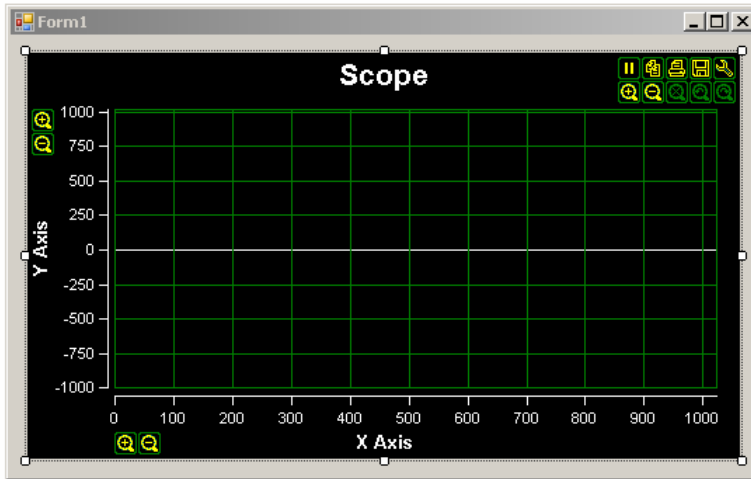
- SignalGen

From the “PlotLab” tab on the Toolbox select and drop on the form the following component:

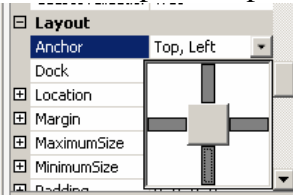


- Scope

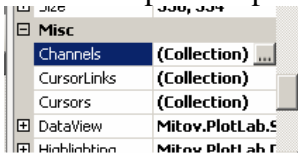
Select the scope1 component on the form editor, and arrange it to look like on this picture:



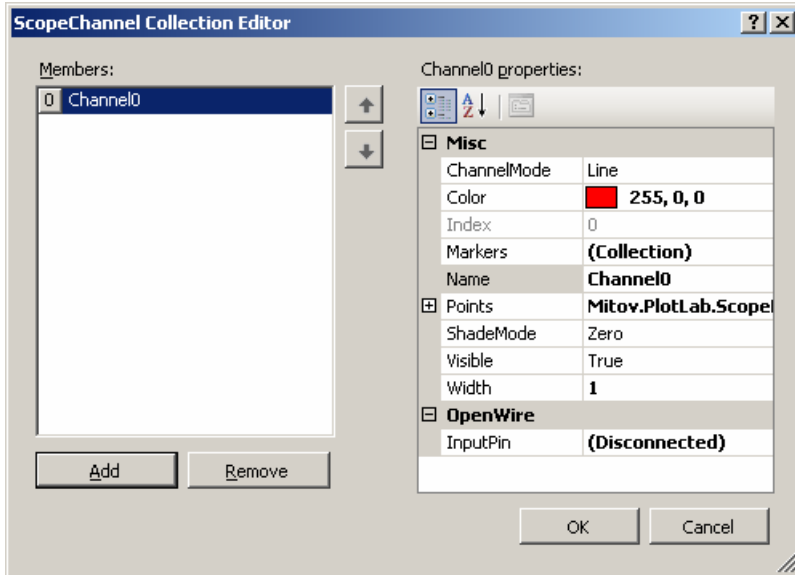
In the “Properties” palette set the Anchor property as shown here:



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

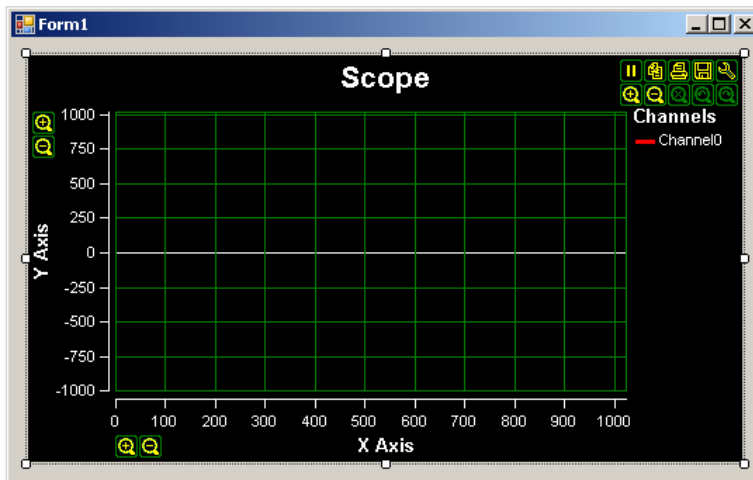


In the “ScopeChannel Collection Editor” add a channel by clicking twice on the “Add” button:



Click the “OK” button.

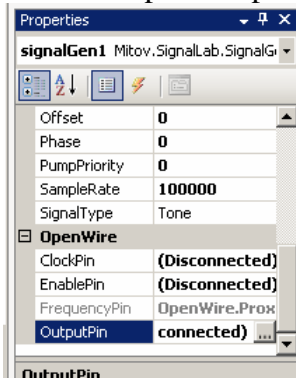
Now your scope component should look similar to this picture:



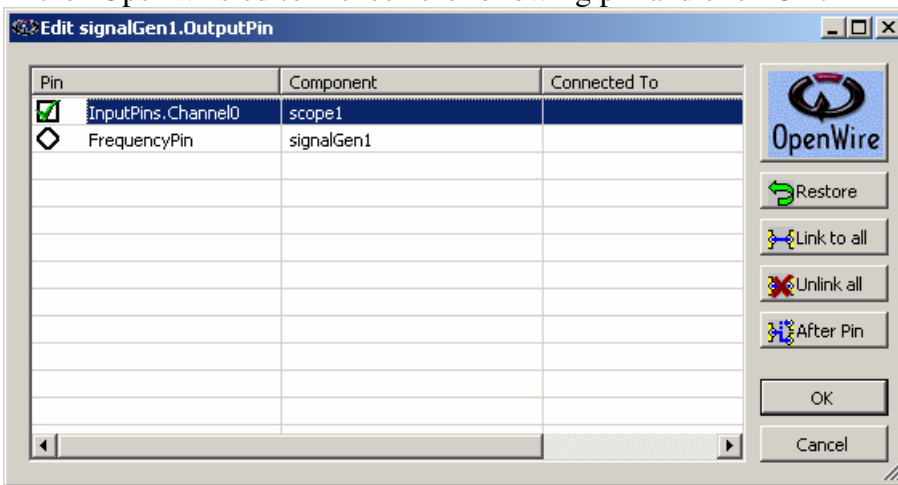
Select the signalGen1 component on the form editor:



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

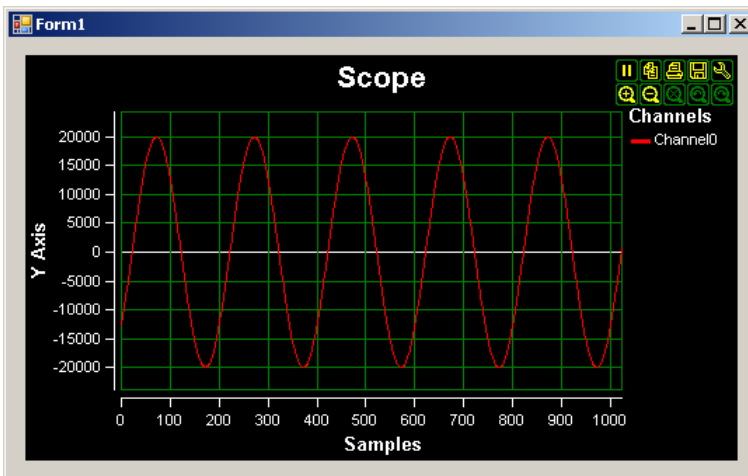


In the “OpenWire editor” check the following pin and click OK:



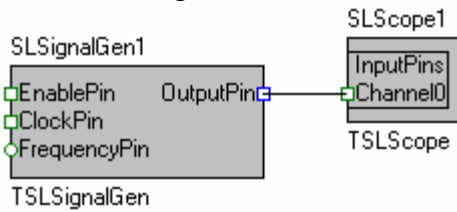
Compile and run the application.

You should see the sine wave:



Congratulations! You have just created your first SignalLab 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.