Antibolab 7,5 NET Quick Start



www.openwire.org www.mitov.com

Copyright Boian Mitov 2004 - 2014

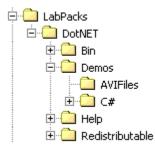
IIIdex	
Installation	3
Where is AudioLab?	3
Creating a new AudioLab project in Visual C#	3
Installing the components on the Toolbox	6
Adding the necessary assembly references to your application	8
Creating a simple video player using DirectShow components	9
Distributing your application	10
Deploying your 32 bit application with the IPP DLLs.	11
Deploying your 64 bit application	11

Installation

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

Where is AudioLab?

After the installation AudioLab 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:

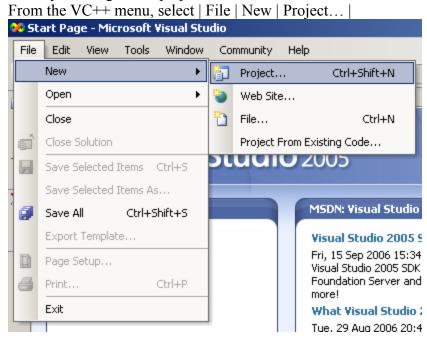


Under the "Demos" directory are located the demo files. The help files and the documentation are located under the "Help" directory. The component .NET 2.0/3.5/4.0 assemblies and the redistributable DLL files are located under the "Bin" directory. It is a great idea to start by opening and compiling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

Creating a new AudioLab 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 AudioLab functionality.

Start by creating a new project.



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

			? >
	<u>T</u> emplates:		000 000 000 000 000
rice s Types	Windows Application Windows Control Library Console Application Empty Project	Glass Library 행 Web Control Library 행 Windows Service 행 Crystal Reports Application	
ting an application v	vith a Windows user interface		
WindowsApplicat	ion1		
C:\Documents ar	nd Settings\Boian Mitov\My Documents\V	isual Studio 2005\Projects 🔹 💌	Browse
WindowsApplicat	ion1	Create directory for solution	
		ОК	Cancel
	s Types ting an application v WindowsApplicat C:\Documents ar	ice Windows Application Windows Control Library Console Application Empty Project S Types ting an application with a Windows user interface WindowsApplication1 C:\Documents and Settings\Boian Mitov\My Documents\W	Windows Application Class Library Windows Control Library Web Control Library Console Application Windows Service Empty Project Crystal Reports Application s Search Online Templates Types Search Online Templates ting an application with a Windows user interface WindowsApplication1 C:\Documents and Settings\Boian Mitov\My Documents\Visual Studio 2005\Projects WindowsApplication1 C:\Documents and Settings\Boian Mitov\My Documents\Visual Studio 2005\Projects

Click OK.

Installing the components on the Toolbox

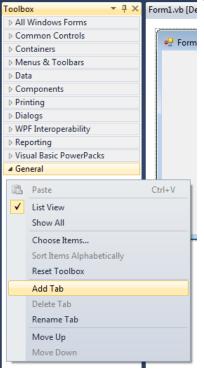
Before using the components in your project, you will have to install them on the component Toolbox.

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:

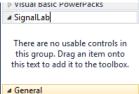
We assume that you have already created a project, and the toolbox with the .NET components has appeared.

Open the component toolbox and expand the General section.

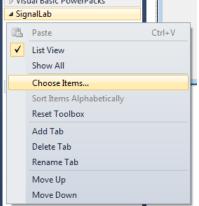
Right-click and select |Add Tab| from the menu:



Name the new tab "SignalLab":



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

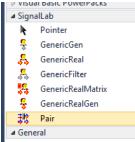


In the "Choose Toolbox Items" dialog select the components that belong to the Mitov.SignalLabBasic.DLL (You can order the components by "Assembly Name"). Press the "Spacebar" key to make sure all the components are selected:

Silverlight Components	s Systen	n.Workflow Components		System.Activities	s Components	;
.NET Framework Co	mponents	COM Components		WPF C	omponents	
Name	Namespace		Assembl	y Name	Directory	*
GenericFilter	Mitov.SignalLab		Mitov.Si	gnalLabBasic	Global Asse.	
🛛 GenericGen	Mitov.SignalLab		Mitov.Si	- gnalLabBasic	Global Asse.	
🗹 GenericReal	Mitov.SignalLab		Mitov.Si	gnalLabBasic	Global Asse.	
🗹 GenericRealGen	Mitov.SignalLab		Mitov.Si	gnalLabBasic	Global Asse.	-
🛛 GenericRealMatrix	Mitov.SignalLab		Mitov.Si	gnalLabBasic	Global Asse.	
🔽 Pair	Mitov.SignalLab		Mitov.Si	gnalLabBasic	Global Asse.	
BackgroundWorker	System.Compon	entModel	System		Global Asse.	
Component	System.Compon	entModel	System		Global Asse.	
EventLog	System.Diagnost	ics	System		Global Asse.	
FileSystemWatcher	System.IO	1	System		Global Asse.	
MarchalBul/alusCom	Sustam Company	entModel	Surtom		Global Acco	Ĩ
<u>F</u> ilter:					<u>C</u> lear	
– GenericFilter 🐥 Language: In	variant Language ()	Invariant Country)			<u>B</u> rowse	
			ок	Cancel	Res	-+

Click OK.

You should see the SignalLabBasic components on your toolbox:



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

Repeat the same steps adding 4 more tabs named "PlotLab", "TimingLab", "MediaLab", and "AudioLab".

On the "PlotLab" tab install Mitov.PlotLabBasic.dll. On the "TimingLab" tab install Mitov.TimingLabBasic.dll. On the "MediaLab" tab install Mitov.MediaLabBasic.dll. On the "AudioLab" tab install Mitov.AudioLabBasic.dll, Mitov.AudioLabEssential.dll and Mitov.AudioLab.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.AudioLabBasic.dll from the LabPacks\Bin\2.0 or 4.0 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.SignalLabAdditional.DLL: a. Mitov.BasicLab.DLL
- For Mitov.TimingLabBasic.DLL:

- a. Mitov.BasicLab.DLL
- For Mitov.PlotLabBasic.DLL:
 - a. Mitov.BasicLab.DLL
- For Mitov.MediaLabBasic.DLL:
 - a. Mitov.BasicLab.DLL
 - b. Mitov.AudioLabBasic.DLL
- For Mitov.AudioLabBasic.DLL:
 - a. Mitov.BasicLab.DLL
- For Mitov.AudioLabEssential.DLL:
 - a. Mitov.AudioLabBasic.DLL
 - b. Mitov.BasicLab.DLL
- For Mitov.AudioLab.DLL:
 - a. Mitov.AudioLabEssential.DLL
 - b. Mitov.AudioLabBasic.DLL
 - c. Mitov.BasicLab.DLL

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 recommend checking to make sure your system is up to date.

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

From the "AudioLab" tab on the Toolbox select and drop on the form the following two components:

DSAudioOut
DSAudioPlayer

Select the dsAudioPlayer1 component on the form editor:



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



Edit dsAudioPlaye			
Pin	Component	Connected To	673
📶 InputPin	dsAudioOut1		OpenWir
			<mark>}-</mark> ∳Link to all
			💥 Unlink all
			🥌 After Pin
			ОК
•1	i.		Cancel

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

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

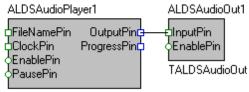
Pr	operties	+ + ×
d٩	AudioPlayer1	Mitov, AudioLab, DS 🗸
	2↓ 🗉 🖋	
Ξ	Misc	_
	BuffersAhead	0
	ClockSource	Internal
	Enabled	True
	FileName	iles\Demo.wav
Ð	Graph	Mitov.AudioLat
	Loop	False
	Paused	False
	PumpPriority	0

Compile and run the application.

You should see the form and hear the audio playing:



Congratulations! You have just created your first AudioLab DirectShow application. Here are the OpenWire connections in this application:



TALDSAudioPlayer

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.

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.