

Visual C++ Quick Start



www.openwire.org www.mitov.com

Copyright Boian Mitov 2004 - 2011

Index

3
3
3
.17
.21
.31
.33
.34
•

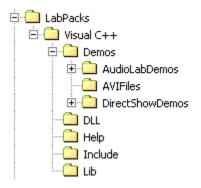
Installation

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. 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 AudioLabDemos and DirectShowDemos directories are located the demo files. The help files and the documentation are located under the Help directory. The DLL directory contains the redistributable DLL files. The header files needed for your projects are located under the Include directory. The Release and Debug version of the library is located under the Lib directory.

It is a great idea to start by opening and compiling the demo files. The demo projects ware designed with Visual C++ 6.0. They can be opened and compiled under Visual C++.NET as well, in this case the IDE will create the necessary solution files.

Creating a new AudioLab project in Visual C++

All of the examples in this manual start with creating a MFC Dialog based project. This is not a AudioLab requirement, but using the resource editor to design the application makes writing the examples much easier.

The following chapters will assume that you have created the project and will teach you how to add specific AudioLab functionality.

Visual C++ 6.0:

v Isual C++ 0.0.	
Start by creating a new project.	
From the VC++ menu, select File New	
Microsoft Visual C++	
File Edit View Insert Project	
New Ctrl+N	
Open Workspace	
Sa <u>v</u> e Workspace	
Close Wor <u>k</u> space	
Save Ctrl+5	
Save As	
Save All	
- Dave Hi	
Page Setyp	
Brint Ctrl+P	
Recent <u>Files</u>	
Recent Workspaces	
Exit	
The project type dialog will appear. Select t	he MFC AppWizard:
New	? 🗙
Files Projects Workspaces Other Documents	
ATL COM AppWizard	Project name:
Cluster Resource Type Wizard	
Custom AppWizard	
a Database Project	Location:
🕸 DevStudio Add-in Wizard	C:\MyTreasureLabProjects\
🖆 Extended Stored Proc Wizard	
ISAPI Extension Wizard	
	Create new workspace
Image: Image	C Add to current workspace
MFC AppWizard (exe)	Dependency of:
T Utility Project	<u> </u>
Win32 Application	
Win32 Console Application	Platforms:
📽 Win32 Dynamic-Link Library	Platroms: ♥Win32
🔊 Win32 Static Library	▼ ₩ III JZ
<u>, , , , , , , , , , , , , , , , , , , </u>	
<u></u>	OK Cancel
	OK. Cancel

Type a project name. For each example the	project name will be different:
New	? 🗙 ?
Files Projects Workspaces Other Documents	
 ATL COM AppWizard Cluster Resource Type Wizard Custom AppWizard Database Project DevStudio Add-in Wizard Extended Stored Proc Wizard ISAPI Extension Wizard ISAPI Extension Wizard MFC ActiveX ControlWizard MFC AppWizard (dll) MFC AppWizard (exe) Utility Project Win32 Application Win32 Dynamic-Link Library Win32 Static Library 	Project <u>name:</u> MyFirstProject Logation: C:\MyTreasureLabProjects\MyFi C Create new workspace Add to current workspace Dependency of: Platforms: Win32
	OK Cancel
Click OK.	
Select a Dialog base project from Step 1 and	d click Next
MFC AppWizard - Step 1	
Application OK OK Cascel Multiple documents Dialog based Document/View arch What language would you l English [United States] (ould you like to create? nitecture support? like your resources in?
< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cancel

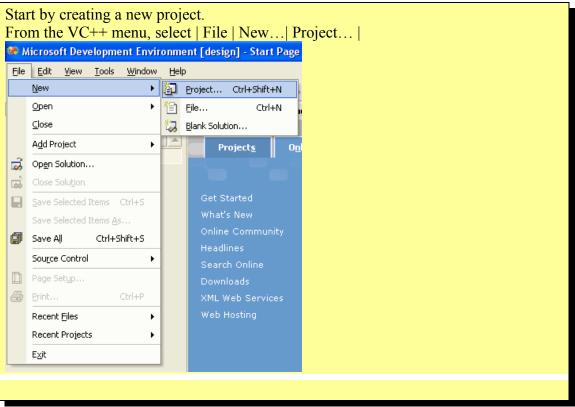
For simplicity disable the	ActiveX Controls on Step 2 and c	lick Next:
MFC AppWizard - Step 2 of 4	? 🗙	
	What features would you like to include?	
Application		
Cancel	Context-sensitive H <u>e</u> lp	
	✓ <u>3</u> D controls W/het attact are accorded up to like to include 2.	
	What other support would you like to include?	
	Automation ActiveX Controls	
	Would you like to include WOSA support?	
Editing Control: Record	_	
Check Box 💿 Radio Button	Windows Sockets	
	Please enter a <u>t</u> itle for your dialog:	
	SimpleVideoPlayer	
< <u>B</u> ack	<u>N</u> ext > <u>F</u> inish Cancel	
T 1 1 0 1		
Leave the default options		
MFC AppWizard - Step 3 of 4	? 🔀	
Microsoft Developer Studio	What style of project would you like ?	
File Edit Yiew Insert Build Help	MFC Standard	
Project Project.cpp #-♥ // TOD0:	C Windows Explorer	
	Would you like to generate source file comments?	
	 Yes, please 	
Ready	 No, thank you 	
	How would you like to use the MFC library?	
	• As a shared <u>D</u> LL	
	\bigcirc As a <u>s</u> tatically linked library	
< <u>B</u> ack	<u>N</u> ext > <u>F</u> inish Cancel	

MEC AppWizard - Step 4 of 4 AppWizard creates the following glasses for you Multiplication glasses for you Multiplication ChyfirstProject.Dog Cless name: Header file: ChyfirstProject.App Multiplication file: ChyfirstProject.App Multiplication file: ChyfirstProject.cpp Reset Reset Enrich Cancel Confirm the selection by clicking OK: Multiplication point of SimpleVideoReget AppWizard will create a new schetor project with the following specification: Paper SimpleVideoReget Paper SimpleVideoRe	Click Finish on step 4:			_
Image: Control of Contro	MFC AppWizard - Step 4 of 4		? 🛛	
CMyFirstProjectApp MyFirstProject h Base class: Implementation file: CWinApp MyFirstProject.cpp MyFirstProject.cpp Confirm the selection by clicking OK: New Project Information AppWirad will create a new skeleton project with the following specifications: AppWirad will create a new skeleton project with the following specifications: AppWirad will create a new skeleton project with the following specifications: Dialog Sead Application targeting: Win32 Classes to be created Application: CSimpleVideoPlayed.pp in SimpleVideoPlayer.h and SimpleVideoPlayed.pp in SimpleVideoPlayer.Bland SimpleVideoPlayed.pp in SimpleVideoPlayer.Bland SimpleVideoPlayed.pp in SimpleVideoPlayer.Bland SimpleVideoPlayed.pp in SimpleVideoPlayer.Bland SimpleVideoPlayed.pp in SimpleVideoPlayer.Bland SimpleVideoPlayed.B		CMyFirstProjectApp	following <u>c</u> lasses for you:	
New Project Information Image: Comparison of the project with the following specifications: Application type of SimpleVideoPlayer: Dialog-Based Application targeting: Win32 Classes to be created: Application: CSimpleVideoPlayerApp in SimpleVideoPlayer.h and SimpleVideoPlayerDlg in SimpleVideoPlayer.h and SimpleVideoPlayerDlg.cpp Features: + 3D Controls + Uses shared DLL implementation (MFC42.DLL) + Localizable text in: English [United States] Project Directory: English [United States]	< <u>B</u> ack	CMyFirstProjectApp Base class: CWinApp	MyFirstProject.h Implementation file: MyFirstProject.cpp	
C:\MyTreasureLabProjects\SimpleVideoPlayer	New Project Information AppWizard will create a new skeleton project Dialog-Based Application targeting: Win32 Classes to be created: Application: CSimpleVideoPlayerApp in Si SimpleVideoPlayer.cpp Dialog: CSimpleVideoPlayerDlg in SimpleV SimpleVideoPlayerDlg.cpp Features: + 3D Controls + Uses shared DLL implementation (MFC4 + Localizable text in: English [United States]	t with the following specifications mpleVideoPlayer.h and /ideoPlayerDlg.h and 2.DLL)		

At this point you should have a new project created. From the menu select Project Settings : pject - Microsoft Visual C++ - [MyFirstProject.rc - ID View Insert Project Build Layout Iools Window Help Set Active Project Add To Project Dependencies Settings Alt+F7 Export Makefile Insert Project into Workspace
In the Project Settings dialog select the Link tab and in the ". Switch to the "Input" cathegory. In the "Additional library path:" edit box add the path to the library files. If you have followed the default installation it should be located at C:\Program Files\LabPacks\Visual C++\Lib:
Settings Setings Setings S
Switch to the $ C/C++ $ tab. In the "Additional include directories:" edit box add the path to the header files. If you have followed the default installation they should be located at C:\Program

Files\LabPacks\Visual C++\Include:			
Project Settings			
Settings For: Win32 Debug	General Debug C/C++ Link Resourc(Category: Preprocessor		
	OK Cancel		
Click OK.			
Now you have fully configured	project, and you can start writing the actual code.		

Visual C++ 2003:



New Project Implates: Project Types: Implates: Visual C# Projects Implates: Visual C++ Projects Implates: Win32 General General Implation Dil Setup and Deployment Projects Implation Dil Chter Projects Implation Dil An application that uses the Microsoft Foundation Class Library. Implation Dil Name: <enter name=""> Implation Location: C:\MyTreasureLabProjects Implation Project will be created at C:\MyTreasureLabProjects\<enter name="">. Implation Implation: OK Cancel Help</enter></enter>
Visual C# Projects Visual C++ Projects ATL ATL MFC Win32 General Other Projects Other Projects MEC ISAPI Extension Dil An application that uses the Microsoft Foundation Class Library. Name: Location: C:\MyTreasureLabProjects Project will be created at C:\MyTreasureLabProjects\ <enter name="">.</enter>
Wisual C++ Projects IFC IFC MFC ATL MFC ATL MFC MFC DLL MFC Win32 Image: An application that uses the Microsoft Foundation Class Library. MFC ISAPI Image: An application that uses the Microsoft Foundation Class Library. Name: <enter name=""> Image: C:\MyTreasureLabProjects Image: Browse Project will be created at C:\MyTreasureLabProjects Image: Seture C:\MyTreasureLabProjects Image: Seture C:\MyTreasureLabProjects</enter>
An application that uses the Microsoft Foundation Class Library. Name: <enter name=""> Location: C:\MyTreasureLabProjects Project will be created at C:\MyTreasureLabProjects\<enter name="">.</enter></enter>
Location: C:\MyTreasureLabProjects Project will be created at C:\MyTreasureLabProjects\ <enter name="">.</enter>
Project will be created at C:\MyTreasureLabProjects\ <enter name="">.</enter>
▼More OK Cancel Help
Type a project name. For each example the project name will be different: New Project
Project Types: Templates:
Visual C# Projects Visual C++ Projects Visual C++ Projects NET ATL MFC MFC MFC ActiveX Control Application MFC DLL MFC
An application that uses the Microsoft Foundation Class Library.
Name: MyFirstProject
Location: C:\MyTreasureLabProjects Project will be created at C:\MyTreasureLabProjects\MyFirstProject.
The sector will be checked at each of the sector of the
Click OK.

-10-

Select a Dialog base pr	oject from Step 1 and c	click Next:
MFC Application Wizard - My	FirstProject	
Application Type Specify Document/View architect application.	ure support, language, and interface s	style options for your
Overview Application Type Compound Document Support Document Template Strings Database Support User Interface Features Advanced Features Generated Classes	Application type: Single document Multiple documents Single documents Multiple top-level documents Multiple top-level documents C Multiple top-level documents Resource language: English (United States)	Project style: Windows Explorer MFC standard Use of MFC: Use MFC in a shared DLL Use MFC in a static library port Finish Cancel Help
E 1 1 4 1 11 4		n Step 2 and click Next:
MFC Application Wizard - My Advanced Features Specify additional support to buil	FirstProject	
Overview Application Type Compound Document Support Document Template Strings Database Support User Interface Features Advanced Features Generated Classes	Advanced features: Context-sensitive Help WinHelp Format HTML Help format Automation Automation ActiveX controls MAPI (Messaging API) Windows sockets Active Accessibility Common Control Manifest	Number of files on recent file list:
Click Finish.		

	Project Settings : esign] - My s Window Shift+A +Alt+A	
library directories:" e	dit box add the path t should be located at (ges Platform: Active Output File Show Progress Version Enable Incremental Linking Suppress Startup Banner Ignore Import Library Register Output Additional Library Directories	\$(OutDir)/\$(ProjectName).exe Not Set Yes (/INCREMENTAL) No No No C:\Program Files\LabPacks\Visual C++\Lib
	semi-colon delimited list if more	than one. (/LIBPATH:[dir])
		OK Cancel Apply Help
	lude directories:" edi	it box add the path to the header files. If you should be located at C:\Program

Files\LabPacks\Visual	l C++\Include:	
MyFirstProject Property Pa	ges	
Configuration: Active(Debug)	Platform: Active(Win3	2) <u>Co</u> nfiguration Manager
🚖 Configuration Propertie 🔼	Additional Include Directories	C:\Program Files\LabPacks\Visual C++\I
General	Resolve #using References	
Debugging	Debug Information Format	Program Database for Edit & Continue (/ZI)
🔄 C/C++	Suppress Startup Banner	Yes (/nologo)
💠 General	Warning Level	Level 3 (/W3)
Optimization	Detect 64-bit Portability Issues	Yes (/Wp64)
Preprocessor Code Generatio	Treat Warnings As Errors	No
Language Precompiled He- Output Files Browse Informa Advanced Command Line Linker Resources Browse Information Build Events Custom Build Step	Additional Include Directories Specifies one or more directories to list if more than one. (/I[path])	add to the include path; use semi-colon delimited
	ОК	Cancel <u>Apply</u> Help
Click OK.		
Now you have fully co	onfigured project, and y	ou can start writing the actual code

Creating a simple audio player using Win32API Components

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

Select the components on the dialog form:

AudioPlayer		×
	ок]
	Cancel	
TODO: Place dialog controls here.		-0.0

Click the "Del" key. They will be deleted from the form:

Ч		
9	AudioPlayer	X
ļ		
P		

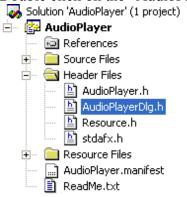
In Visual C++ 6.0: Select the "FileView" tab:

<		
ClassView	🕵 ResourceVi	FileView

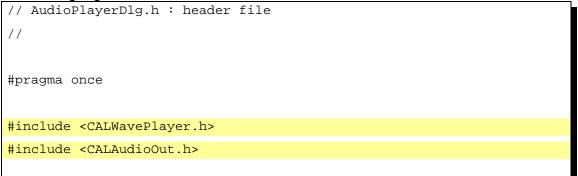
In Visual Visual C++ 2003/2005: Select the "Solution Explorer" tab:

Solution Explorer	<u> </u>	Class View	📳 Res

Double click on the "AudioPlayerDlg.h":



Add the highlighted lines in the header:



```
// CAudioPlayerDlg dialog
class CAudioPlayerDlg : public CDialog
{
// Construction
public:
     CAudioPlayerDlg(CWnd* pParent = NULL); // standard
constructor
// Dialog Data
     enum { IDD = IDD_AUDIOPLAYER_DIALOG };
    protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV
support
// Implementation
protected:
     CTALWavePlayer
                     ALWavePlayer;
     CTALAudioOut
                     ALAudioOut;
protected:
    HICON m_hIcon;
     // Generated message map functions
     virtual BOOL OnInitDialog();
     afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
     afx_msg void OnPaint();
     afx_msg HCURSOR OnQueryDragIcon();
     DECLARE MESSAGE MAP()
};
```

Double click on the "AudioPlayerDlg.cpp" file:

Solution 'AudioPlayer' (1 project)

AudioPlayer

AudioPlayer

AudioPlayer.cpp

AudioPlayerDlg.cpp

Header Files

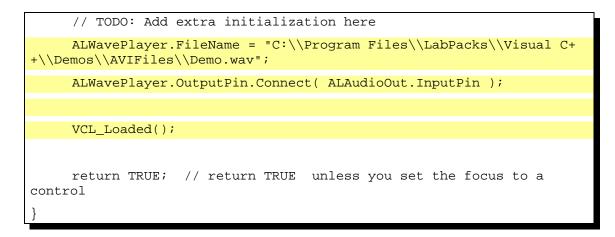
AudioPlayer.manifest

ReadMe.txt

Add the highlighted lines in the CAudioPlayerDlg::OnInitDialog method:

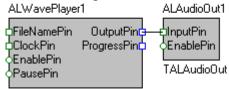
```
BOOL CAudioPlayerDlg::OnInitDialog()
{
     CDialog::OnInitDialog();
     // Add "About..." menu item to system menu.
     // IDM_ABOUTBOX must be in the system command range.
     ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
     ASSERT(IDM ABOUTBOX < 0xF000);
     CMenu* pSysMenu = GetSystemMenu(FALSE);
     if (pSysMenu != NULL)
     {
           CString strAboutMenu;
           strAboutMenu.LoadString(IDS_ABOUTBOX);
           if (!strAboutMenu.IsEmpty())
           {
                 pSysMenu->AppendMenu(MF_SEPARATOR);
                 pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX,
strAboutMenu);
           }
     }
     // Set the icon for this dialog. The framework does this
automatically
     // when the application's main window is not a dialog
     SetIcon(m hIcon, TRUE);
                                         // Set big icon
     SetIcon(m_hIcon, FALSE);
                                         // Set small icon
```

October 11, 2011



Compile and run the application. You should hear the wave file playing.

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



TALWavePlayer

Creating a simple audio player using DirectShow components

WARNING: In order to run the application in this example you must have DirectX 9.0 or higher installed!

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

🗖 AudioPlayer OK Cancel TODO: Place dialog controls here.

Select the components on the dialog form:

October 11, 2011

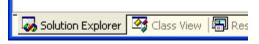
Click the "Del" key. They will be deleted from the form:



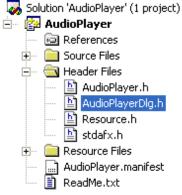
In Visual C++ 6.0: Select the "FileView" tab:



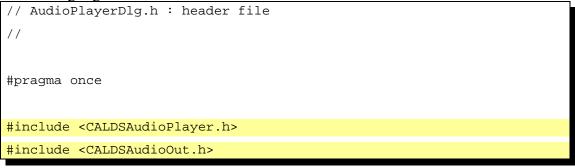
In Visual Visual C++ 2003/2005: Select the "Solution Explorer" tab:



Double click on the "AudioPlayerDlg.h":



Add the highlighted lines in the header:



October 11, 2011

```
// CAudioPlayerDlg dialog
class CAudioPlayerDlg : public CDialog
{
// Construction
public:
     CAudioPlayerDlg(CWnd* pParent = NULL); // standard
constructor
// Dialog Data
     enum { IDD = IDD_AUDIOPLAYER_DIALOG };
     protected:
     virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV
support
// Implementation
protected:
     CTALDSAudioPlayer ALDSAudioPlayer;
     CTALDSAudioOut ALDSAudioOut;
protected:
     HICON m_hIcon;
     // Generated message map functions
     virtual BOOL OnInitDialog();
     afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
     afx_msg void OnPaint();
     afx_msg HCURSOR OnQueryDragIcon();
     DECLARE_MESSAGE_MAP()
```

Double click on the "AudioPlayerDlg.cpp" file:

Solution 'AudioPlayer' (1 project)

AudioPlayer

AudioPlayer

AudioPlayer.cpp

AudioPlayerDlg.cpp

Header Files

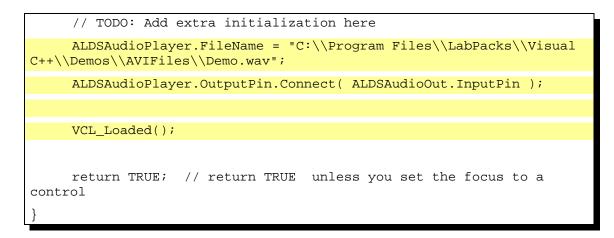
AudioPlayer.manifest

ReadMe.txt

Add the highlighted lines in the CAudioPlayerDlg::OnInitDialog method:

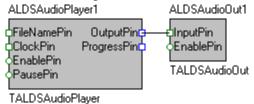
```
BOOL CAudioPlayerDlg::OnInitDialog()
{
     CDialog::OnInitDialog();
     // Add "About..." menu item to system menu.
     // IDM_ABOUTBOX must be in the system command range.
     ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
     ASSERT(IDM ABOUTBOX < 0xF000);
     CMenu* pSysMenu = GetSystemMenu(FALSE);
     if (pSysMenu != NULL)
     {
           CString strAboutMenu;
           strAboutMenu.LoadString(IDS_ABOUTBOX);
           if (!strAboutMenu.IsEmpty())
           {
                 pSysMenu->AppendMenu(MF_SEPARATOR);
                 pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX,
strAboutMenu);
           }
     }
     // Set the icon for this dialog. The framework does this
automatically
     // when the application's main window is not a dialog
     SetIcon(m hIcon, TRUE);
                                         // Set big icon
                                         // Set small icon
     SetIcon(m_hIcon, FALSE);
```

October 11, 2011



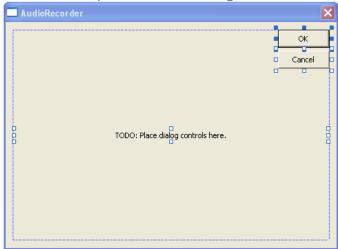
Compile and run the application. You should hear the wave file playing.

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



Creating a simple Audio Recorder Win32API Components

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



Select the components on the dialog form:

Click the "Del" key. They will be deleted from the form:

	llllllll	
	AudioRecorder	
	-	
E		
1:		
1:		
13		
:		
14		
13		
14		
13		
2		
E		
Ŀ		
_		

From the controls toolbar select a "Static Text" control: VC++ 6 VC2003/2005

V C I I	0.	VC2003/2003.		
Con 🔀		Toolbox	- # 🗙	С
▶ ∰		Dialog Editor		
Aa abl		Radio Button		
רם נ ^{ייא} ן		A α Static Text		
		😥 Picture Control		F
		🕩 Horizontal Scroll Bar		

Place the control on the form:

AudioRecorder		
Itatic		

In Visual C++ 6.0:

From the menu select View Properties : SignalGenerator - Microsoft Visual C+ ClassWizard Ctrl+W Resource Symbols Resource Includes Fyll Screen Workspace Alk+0 Qutput Alk+2 Debug Windows Refresh Properties Alk+Enter
Change the control's ID to "IDC_SCOPE": Text Properties Fixed Properties General Styles Extended Styles ID: IDC_SCOPE Caption: Static Visible Group Help ID Disabled Tab stop
Switch to the "Extended Styles" tab and check the "Client edge" property so you can easily see the control on the dialog: Text Properties

In the "ClassWizard" select the "Member Variables" tab and select the "ID_SCOPE" in the "Control IDs" list box, then press "Add Variable": MFC ClassWizard Message Maps Member Variables Automation ActiveX Events Class Info Project: Class game: Add Class Add Class Add Class Class Info C	From the menu select View ClassWizard : SignalGenerator - Microsoft Visual C+ File Edit View Insert Project Build Lavo Presource Symbols Resource Includes Properties Alt+2 Properties Alt+Enter
in the "Control IDs" list box, then press "Add Variable…": MFC ClassWizard Message Maps Member Variables Automation ActiveX Events Class Info Project: Class game: SignalGenerator CSignalGeneratorDlg Add Class Add Class Add Variable Delete Variable Update Columns Bind All	
OK Cancel	in the "Control IDs" list box, then press "Add Variable…": MFC ClassWizard Message Maps Member Variables Automation ActiveX Events Class Info Project: Class game: Add Class GignalGeneratorDlg C: VsignalGeneratorDlg C: VsignalGeneratorDlg.cpp Control [Ds: Type Member Description:

Set the variable "Category" to be "Control", and set the name to be m_Scope:	
Add Member Variable	
Member variable <u>n</u> ame: OK	
m_Scope Cancel	
Category:	
Control	
Variable type:	
CStatic	
Description:	
map to CStatic member	
Click OK.	
In the "ClassWizard" click OK:	
MFC ClassWizard	
Message Maps Member Variables Automation ActiveX Events Class Info	
Project: Class name: Add Class	
SignalGenerator CSignalGeneratorDlg Add Variable	
Control IDs: Type Member Delete Variable	
IDC_SCOPE CStatic m_Scope Update Columns	
Bind All	
Description: map to CStatic member	
OK Cancel	

In Visual C++ 2003/2005:

Change t	he control's ID to	"IDC_SCOPE":
Group	True	
ID	IDC_SCOPE 🗾	
Tabstop	False 🔽	
Properties	2 Dynamic Help	

Set the "Client Edge" pro Properties DDC_STATIC2 (Text Control) IStatEdit DE 21 0 7 0 Center Image False Client Edge True End Ellipsis False Modal Frame False	3	n easily see the control on the dialog:
From the menu select Pro • Microsoft Visual C++ [design] - Filter Project Build Debug Profile Format ** Add Class ** Add Qlass ** Add Qlass ** Add Resource ** Add Resource ** Add New Item ** Add New Item ** Add Existing Item ** Add Existing Item ** Add Wgb Reference ** Set as StartUp Project ** FiltersDemo Properties		:
In the "Add Member Var "Variable name" to be m Add Member Variable Wizard - S Welcome to the Add Member This wizard adds a member variable to	_Scope: iignalGenerator Variable Wizard	Variable type" to CStatic, and set the
Access: public variable type: CStatic Variable name: m_Scope Comment (// notation not required):	✓ Control variable Control ID: IDC_SCOPE Control type: LTEXT Min value:	Category: Control Max chars: Max value: .cpp file:
Click Finish.		

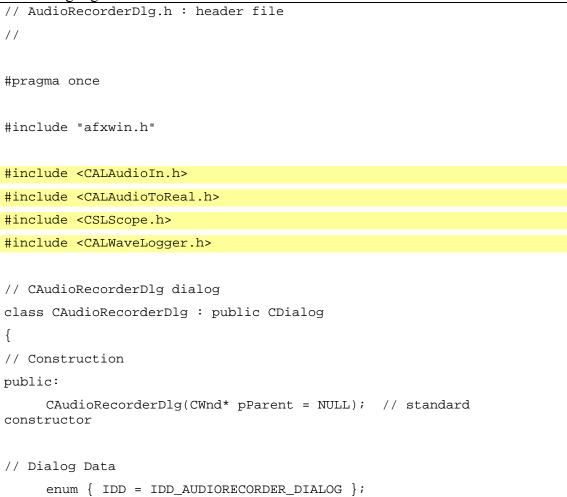
In Visual C++ 6.0: Select the "FileView" tab: In Visual Visual C++ 2003/2005: Select the "Solution Explorer" tab:



Double click on the "AudioRecorderDlg.h":



Add the highlighted lines in the header:



October 11, 2011

<pre>protected: virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support</pre>				
// Implementation				
protected:				
CTALAudioIn	ALAudioIn;			
CTALAudioToReal	ALAudioToReal;			
CTSLScope	SLScope;			
CTALWaveLogger	ALWaveLogger;			
protected: HICON m_hIcon;				
// Generated message map functions				
virtual BOOL OnInitDialog();				
afx_msg void OnSysCommand(UINT nID, LPARAM lParam);				
afx_msg void OnPaint();				
afx_msg HCURSOR OnQueryDragIcon();				
DECLARE_MESSAGE_MAP()				
public:				
CStatic m_Scope;				
};				

Double click on the "AudioRecorderDlg.cpp" file:

Solution 'AudioRecorder' (1 project)

AudioRecorder

AudioRecorder

Source Files

AudioRecorder.cpp

AudioRecorderDlg.cpp

Header Files

AudioRecorder.manifest

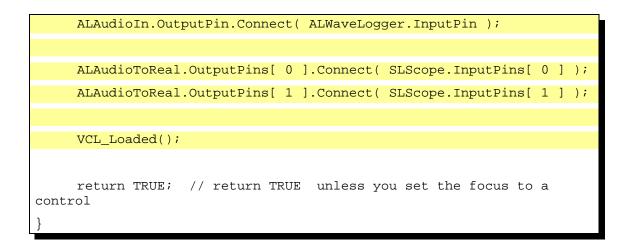
ReadMe.txt

Add the highlighted lines in the CAudioRecorderDlg::OnInitDialog method:

```
BOOL CAudioRecorderDlg::OnInitDialog()
{
     CDialog::OnInitDialog();
```

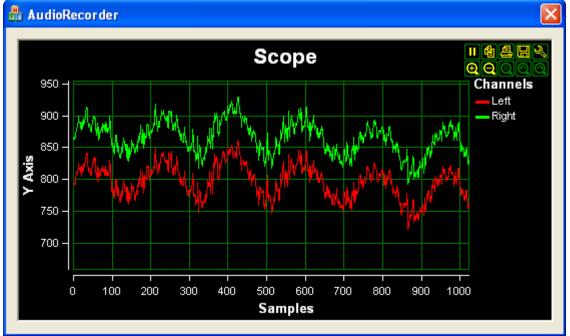
```
// Add "About..." menu item to system menu.
     // IDM_ABOUTBOX must be in the system command range.
     ASSERT((IDM ABOUTBOX & 0xFFF0) == IDM ABOUTBOX);
     ASSERT(IDM_ABOUTBOX < 0xF000);
     CMenu* pSysMenu = GetSystemMenu(FALSE);
     if (pSysMenu != NULL)
     {
           CString strAboutMenu;
           strAboutMenu.LoadString(IDS_ABOUTBOX);
           if (!strAboutMenu.IsEmpty())
           {
                 pSysMenu->AppendMenu(MF_SEPARATOR);
                 pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX,
strAboutMenu);
           }
     }
     // Set the icon for this dialog. The framework does this
automatically
     // when the application's main window is not a dialog
     SetIcon(m_hIcon, TRUE);
                                         // Set big icon
     SetIcon(m_hIcon, FALSE);
                                         // Set small icon
     // TODO: Add extra initialization here
     VCL_InitControls( m_hWnd );
     SLScope.Open( m_Scope.m_hWnd );
     SLScope.Channels.Clear();
     SLScope.Channels.Add( 2 );
     SLScope.Channels[ 0 ].Name = "Left";
     SLScope.Channels[ 1 ].Name = "Right";
     ALWaveLogger.FileName = "RecordedAudio.wav";
     ALAudioIn.OutputPin.Connect( ALAudioToReal.InputPin );
```

October 11, 2011



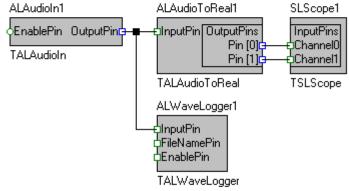
Compile and run the application.

You should see result similar to this one:



A file named RecordedAudio.wav will be created containing the recorded audio.

Here are the OpenWire connections in this application:



You have just learned how to create audio recorder with AudioLab.

Using the TSLCRealBuffer in C++ Builder and Visual C++

The C++ Builder version of the library comes with a powerful data buffer class, called TSLCRealBuffer.

The TSLCRealBuffer is capable of performing basic math operations over the data as well as some basic signal processing functions. The data buffer also uses copy on write algorithm improving dramatically the application performance.

The TSLCRealBuffer is an essential part of the SignalLab generators and filters, but it can be used independently in your code.

You have seen already some examples of using TSLCRealBuffer in the previous chapters. Here we will go into a little bit more details about how TSLCRealBuffer can be used.

In order to use TSLCRealBuffer you must include SLCRealBuffer.h directly or indirectly (trough another include file):

#include <SLCRealBuffer.h>

Once the file is included you can declare a buffer: Here is how you can declare a 1024 samples buffer: TSLCRealBuffer Buffer(1024);

Version 4.0 and up does not require the usage of data access objects. The data objects are now obsolete and have been removed from the library.

You can obtain the current size of a buffer by calling the GetSize method: Int ASize = Buffer.GetSize(); // Obtains the size of the buffers

You can resize (change the size of) a buffer: Buffer.Resize(2048); // Changes the size to 2048

You can set all of the elements (samples) of the buffer to a value: Buffer.Set(30); // Sets all of the elements to 30.

You can access individual elements (samples) in the buffer: Buffer [5] = 3.7; // Sets the fifth elment to 3.7

```
Double AValue = Buffer [ 5 ]; // Assigns the fifth element to a variable
```

You can obtain read, write or modify pointer to the buffer data:

```
const double *data = Buffer.Read() // Starts reading only
double *data = Buffer.Write()// Starts writing only
double *data = Buffer.Modify()// Starts reading and writing
```

October 11, 2011

Sometimes you need a very fast way of accessing the buffer items. In this case, you can obtain a direct pointer to the internal data buffer. The buffer is based on copy on write technology for high performance. The mechanism is encapsulated inside the buffer, so when working with individual items you don't have to worry about it. If you want to access the internal buffer for speed however, you will have to specify up front if you are planning to modify the data or just to read it. The TSLCRealBuffer has 3 methods for accessing the data Read(), Write(), and Modify (). Read() will return a constant pointer to the data. You should use this method when you don't intend to modify the data and just need to read it. If you want to create new data from scratch and don't intend to preserve the existing buffer data, use Write(). If you need to modify the data you should use Modify (). Modify () returns a non constant pointer to the data, but often works slower than Read() or Write(). Here are some examples:

```
const double *pcData = Buffer.Read(); // read only data pointer
```

```
double Value = *pcData; // OK!
```

```
*pcData = 3.5; // Wrong!
```

```
double *pData = Buffer.Write(); // generic data pointer
```

```
double Value = *pData; // OK!
```

```
*pData = 3.5; // OK!
```

You can assign one buffer to another: Buffer1 = Buffer2;

You can do basic buffer arithmetic:

TSLCRealBuffer Buffer1(1024);			
TSLCRealBuffer Buffer2(1024);			
TSLCRealBuffer Buffer3(1024);			
Buffer1.Set(20.5);			
Buffer2.Set(5);			
Buffer3 = Buffer1 + Buffer2;			
Buffer3 = Buffer1 - Buffer2;			
Buffer3 = Buffer1 * Buffer2;			
Buffer3 = Buffer1 / Buffer2;			

In this example the elements of the Buffer3 will be result of the operation (+,-,* or /) between the corresponding elements of Buffer1 and Buffer2.

You can add, subtract, multiply or divide by constant:

// Adds 4.5 to each element of the buffer			
Buffer1 = Buffer2 + 4.5;			
// Subtracts 4.5 to each element of the buffer			
Buffer1 = Buffer2 - 4.5;			
// Multiplies the elements by 4.5			
Buffer1 = Buffer2 * 4.5;			
// Divides the elements by 4.5			
Buffer1 = Buffer2 / 4.5;			

You can do "in place" operations as well:

Buffer1	+=	Buffer2;
Buffer1	+=	4.5;
Buffer1	-=	Buffer2;
Buffer1	-=	4.5;
Buffer1	*=	Buffer2;
Buffer1	*=	4.5;
Buffer1	/=	Buffer2;
Buffer1	/=	4.5;

Those are just some of the basic buffer operations provided by SignalLab.

If you are planning to use some of the more advanced features of TSLCRealBuffer please refer to the online help.

SignalLab also provides TSLCComplexBuffer and TSLCIntegerBuffer. They work similar to the TSLCRealBuffer but are intended to be used with Complex and Integer data. For more information on TSLCComplexBuffer and TSLCIntegerBuffer please refer to the online help.

Distributing your application

Once you have finished the development of your application you most likely will need to distribute it to other systems. In order for the built application to work, you will have to include a set of DLL files together with the distribution. The necessary files can be found under the [install path]\DLL directory([install path] is the location where the library was installed).

You can distribute them to the [Windows]\System32 ([Windows]\SysWOW64 in 64 bit Windows) directory, or to the distribution directory of your application([Windows] is the Windows directory - usually C:\WINNT or C:\WINDOWS).

Deploying your application with the IPP DLLs

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 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.