



Vidview .NET SDK Getting Started Guide

David Hesselbom

November, 2013

Table of Contents

1 Preparations	3
1.1 Introduction	3
1.2 Microsoft Visual Studio	3
1.3 Imint Vidhance License Key	3
1.4 Creating a Visual Studio Project	4
1.5 Setting the Platform Target	5
2 Installing packages	7
2.1 NuGet Packages	7
2.2 Downloading the SDK	7
2.3 Configuring NuGet package manager	7
2.4 Note	11
2.5 Updating packages	11
3 Building	13
3.1 Toolbox	13
3.2 Build and Run	13
3.3 Play a video file	14
4 Using Vidhance	17
4.1 Adding Vidhance Components	17
4.2 Process Separation	18
5 Further exploration	20
5.1 Additional Vidhance Components	20
5.2 Experiment with video overlays	20
5.3 GUI configuration	20
5.4 Live Video	20

1 Preparations

This section will guide you through installing and setting up the tools and utilities you will need in order to use the Vidview .NET SDK.

1.1 Introduction

Imint Vidview is a customizable collection of video viewer applications that can use Imint Vidhance® video enhancements. The Vidview SDK for .NET is an integration toolkit, provided as a Windows Forms control in C# for Microsoft Visual Studio and Microsoft Windows 7. The Vidview SDK makes it possible to embed smart video viewing in a host application. The SDK is provided as a set of NuGet¹ packages, from which a simple Vidview Window standalone application can quickly be built. This application will play video from disk, from capture devices, and video received over network. This Getting Started Guide aims to provide the information needed to get new users up and running with a basic Vidview Window application.

1.2 Microsoft Visual Studio

The Vidview SDK is prepared for Microsoft Visual Studio Express 2012 for Windows Desktop, under the Microsoft .NET Framework version 4.5.50709. Make sure that you have a version of Visual Studio 2012 (or later) installed, and ensure that the installation is updated. If you do not have this installed, you can get a license of Visual Studio Express 2012 from Microsoft.²

Launch Visual Studio, and make sure that its built-in NuGet Package Manager is up to date. If you need help with this step, follow these instructions³ before continuing.

Please note that even a brand new installation of Visual Studio is **not** necessarily up to date.

1.3 Imint Vidhance License Key

If you intend to use functionality from Imint's real-time video enhancement technology, Vidhance, you need to have obtained an appropriate software license key from Imint. If you have not received a license key, please contact Imint. You may also use the license key **novidhance**, which will be accepted as a valid license, but with no Vidhance functionality enabled.

1. <http://nuget.org/>
2. <http://www.microsoft.com/visualstudio/eng/downloads>
3. [http://vidview.imint.se/sdk/net/Updating NuGet.html](http://vidview.imint.se/sdk/net/Updating%20NuGet.html)

1.4 Creating a Visual Studio Project

The following step-by-step instruction aims to guide you through setting up a Visual Studio project for Vidview. The project will build a simple demo application, allowing you to open a video from file and perform basic video control.

1. Launch Visual Studio. In the "File" menu, select "New Project".
2. In the "New Project" dialog, select "Visual C#" templates from the left menu, and select a "Windows Forms Application" as shown below. Give the project a suitable name and location, and press "OK"

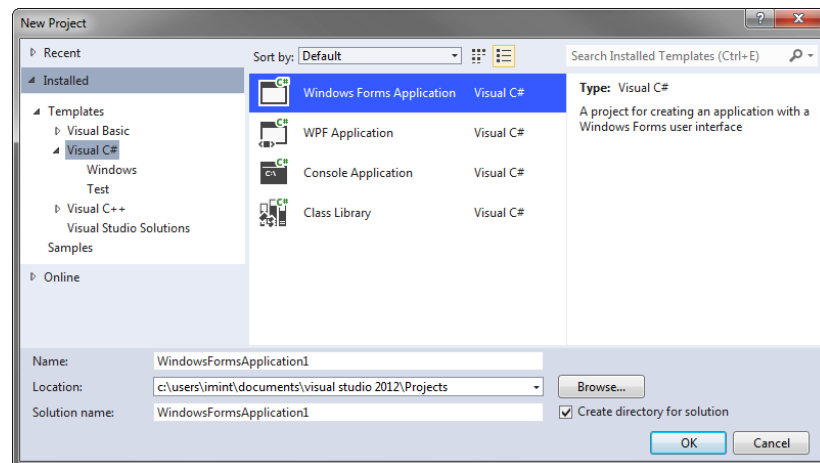


Figure 1: Visual Studio New Project dialog with a Visual C# Windows Forms Application selected.

3. You should now have a project window as shown below.

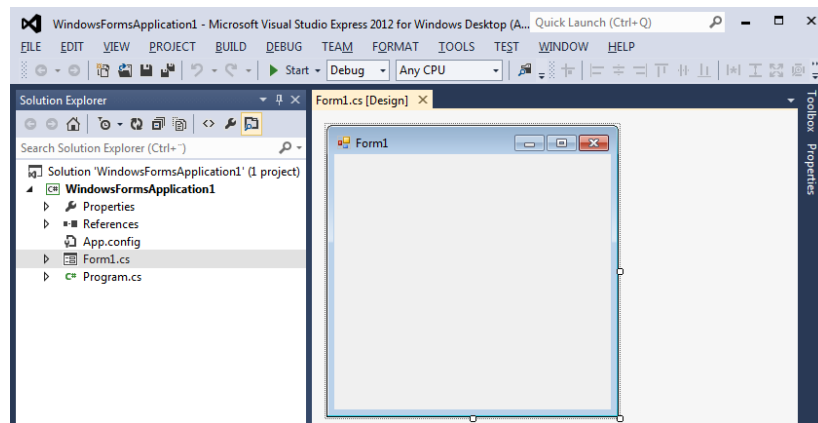


Figure 2: A newly created Visual C# Windows Forms project.

1.5 Setting the Platform Target

By default, Visual Studio projects are set to use the "Any CPU" platform target. This can cause problems on x64 systems, which may load binaries that are incompatible with Vidview.

1. In the "Solution Explorer", right click on your newly created Windows Forms project, and select "Properties".
2. Select the "Build" tab on the left.
3. In the "Configuration" dropdown menu at the top, select "All Configurations".
4. In the "Platform target" dropdown menu near the top, select "x86".

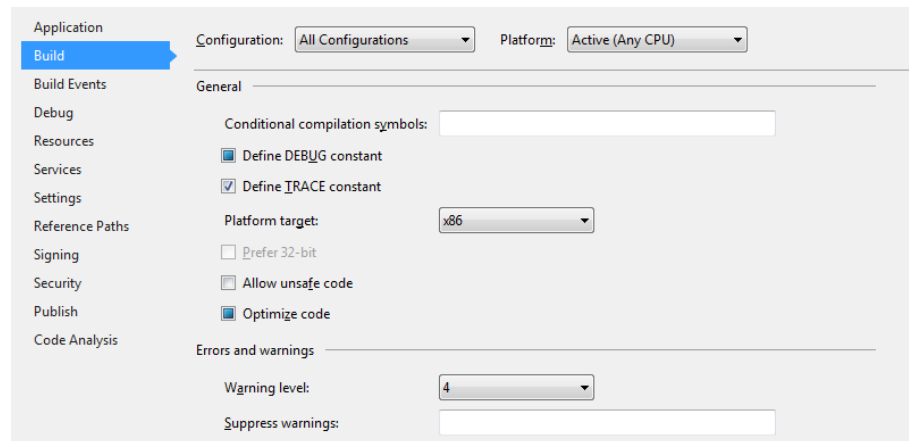


Figure 3: The project properties dialog.

2 Installing packages

2.1 NuGet Packages

The Vidview SDK comes in NuGet packages. NuGet is a package manager that keeps track of dependencies and handles updates. The next few steps will guide you through setting up your own local package feed, containing the Vidview SDK.

2.2 Downloading the SDK

Before you set up a local package feed, download the SDK as a zip archive from <https://aurora.imint.se/download/yourlicensekey/vidview-latest.zip>. Once downloaded, unzip its contents somewhere on disk, e.g. C:\Vidview SDK\. When updates are available, use the same link to download an archive containing the new version.

2.3 Configuring NuGet package manager

2.3.1 Opening the NuGet settings dialog

1. In the "Solution Explorer", right click on your newly created Windows Forms project, and select "Manage NuGet Packages...".
2. In the Manage NuGet Packages dialog, click "Settings".

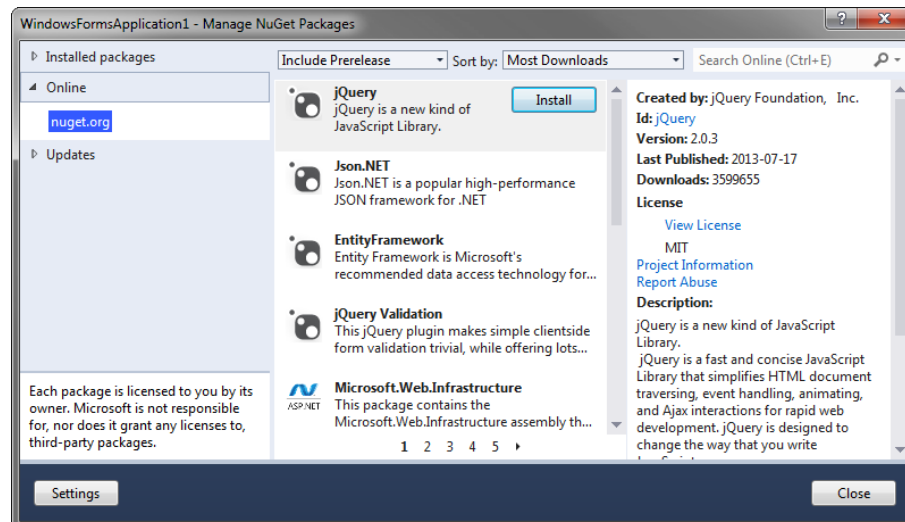


Figure 4: The Manage NuGet Packages dialog with the settings button in the lower left corner.

2.3.2 Adding the package source

Under Package Manager/Package Sources, click the + button to add a new Package Source. Select it in the list and enter the Name `vidview-dev` and Source `C:\Vidview SDK\`. Click Update, then OK.

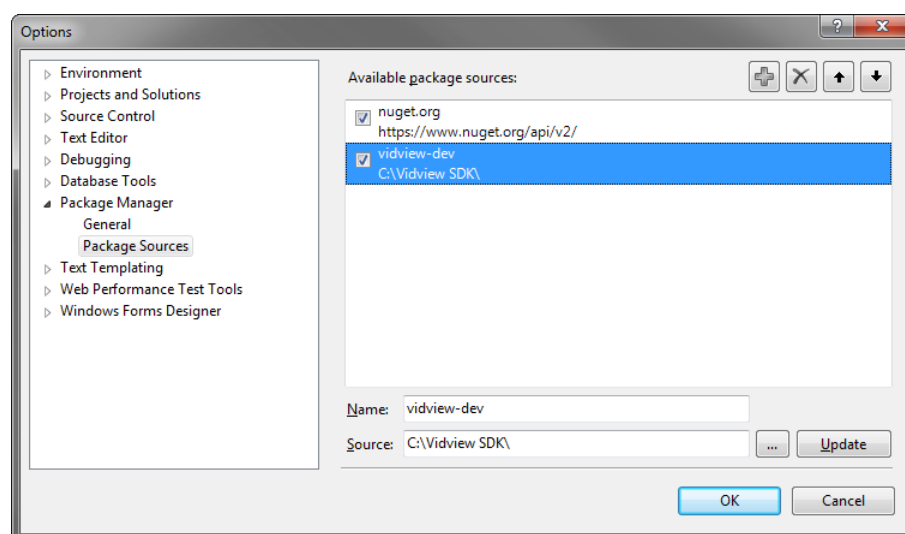


Figure 5: The NuGet settings dialog

2.3.3 Enabling listing of prerelease packages

At the top of the Manage NuGet Packages dialog, there is a drop-down menu with two possible settings:

- Stable Only
- and
- Include Prerelease.

In order to list all of the available packages in the SDK, make sure to select to **Include Prerelease** packages (as shown in Figure 5).

2.3.4 Installing packages

The packages from vidview-dev should now be listed in the Manage NuGet Packages window. It might take a few seconds, but upon completion it should look like below.

You should now install the packages `Imint.Vidview.Sample.Gui` and `Imint.Vidview.Sample.UserControl`. The easiest way to locate these packages is by typing `sample` in the search field on the top right. Once located, click "Install".

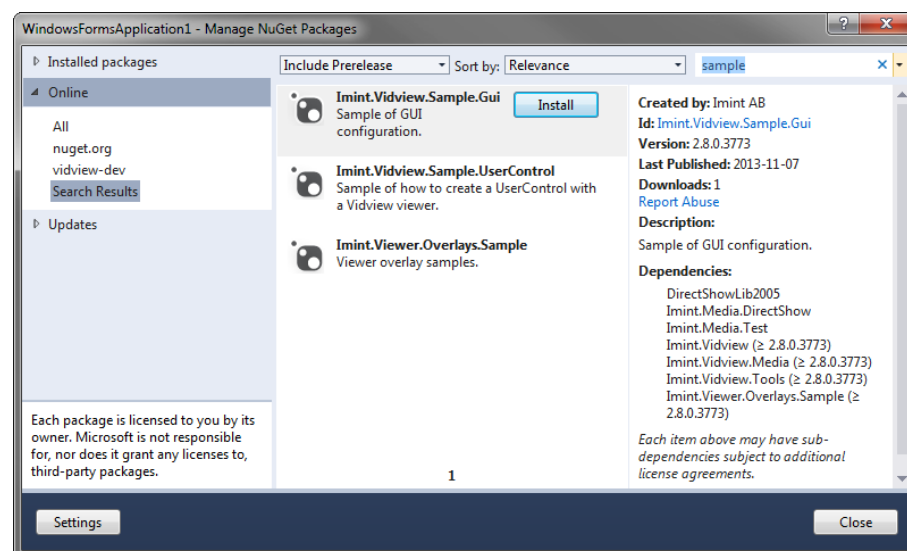


Figure 6: The Manage NuGet Packages dialog showing the `Imint.Vidview.Sample.Gui` package.

The `Imint.Vidview.Sample.Gui` and `Imint.Vidview.Sample.UserControl` packages will automatically install all of their dependencies. This might take a minute, so be patient.

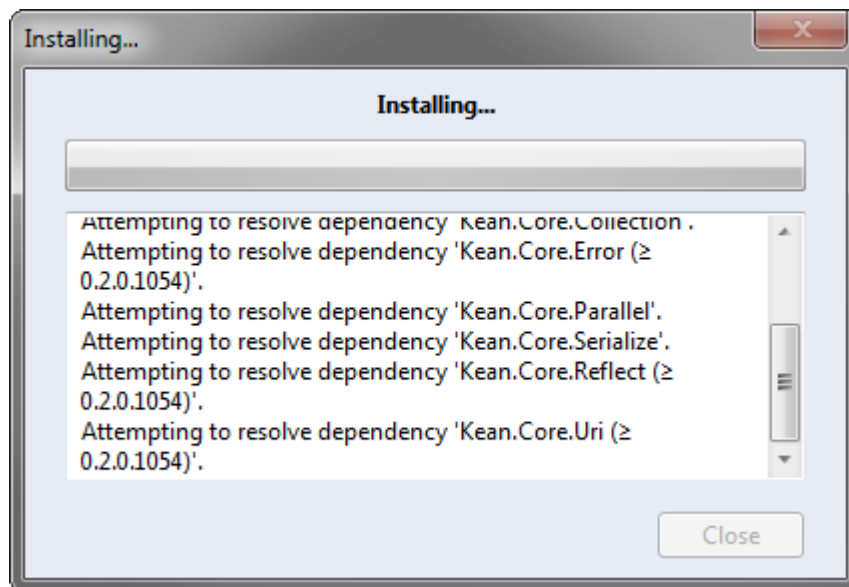


Figure 7: Package installation in progress.

Before closing the Nuget Package Manager window, note "Updates" in the menu to the right. When updates to the Vidview SDK are available, this is where you go to install them. For more information, see the section about [updating packages](#) below.

After installing the packages, build the project, by selecting "Build Solution" from the "Build" menu, or by pressing F7. When ready, the output window should indicate a successful build.

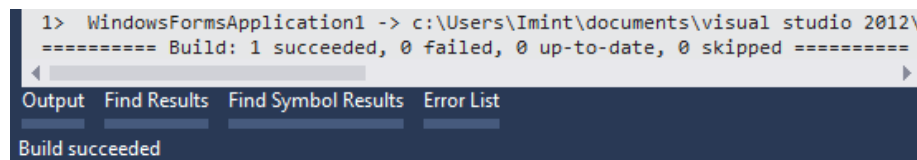


Figure 8: Visual Studio indicating a successful build.

2.4 Note

After you have created a Vidview project, you may wish to use it in another solution. To do so, you can import the Vidview project into your solution and add a reference to it in your own project. However, even after referencing the Vidview project, Visual Studio will fail to copy some DLLs when building your solution. To work around this issue, you must either:

1. change the output path of the Vidview project to be the same as the output path of your other project, or
2. install the NuGet package `Imint.Vidview.Window` in your own project.

2.5 Updating packages

When updates to the SDK are available, you have 3 options:

1. Update the packages,
2. Uninstall the packages in your current project and install new ones, or
3. Start over with a new project.

Updating the packages can be done as follows:

1. You should already have a local feed pointing to a folder where you unzipped the contents of `vidview-dev.zip`, e.g. `C:\Vidview SDK\`. Delete any files in this folder, but keep the folder itself.
2. As described [earlier](#), download the SDK as a zip archive from <https://aurora.imint.se/download/yourlicensekey/vidview-latest.zip>. Once downloaded, unzip its contents into the same folder you did previously, e.g. `C:\Vidview SDK\`.
3. In the "Solution Explorer", right click on your project, and select "Manage NuGet Packages...".
4. In the Nuget Package Manager, select the "Updates" tab on the right.
5. Select the feed `vidview-dev` and allow it to load. Remember to "include prerelease" packages in the top drop-down menu.
6. Once the package manager has finished loading, click the "Update all" button on the top right of the package list to update all installed packages. This will take a few minutes.

Please **note** that there may also be package updates available in the [nuget.org](https://www.nuget.org) feed on the right. Do not update these packages, as the Vidview SDK is dependent on specific versions of these libraries, and updating them may cause version conflicts.

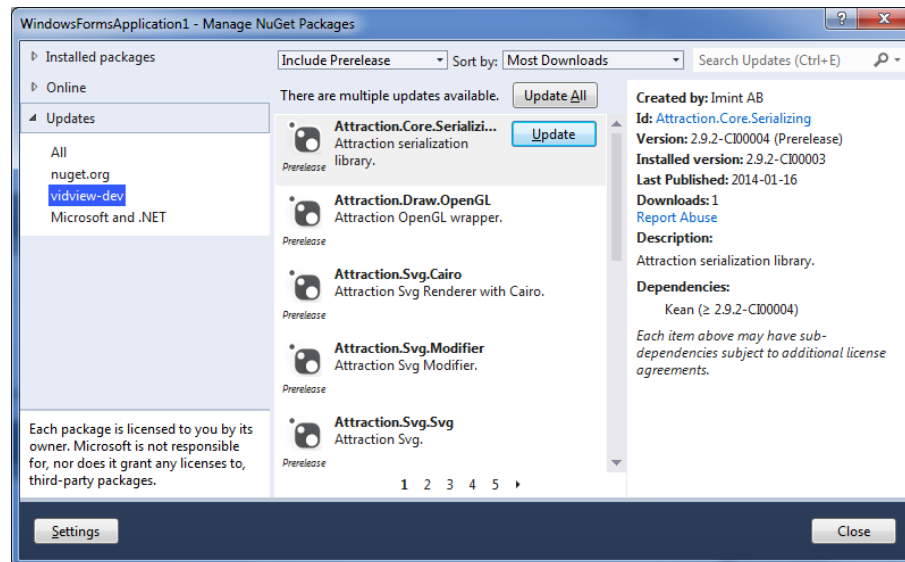


Figure 9: The Updates tab of the NuGet Package Manager.

3 Building

3.1 Toolbox

In the Solution Explorer, double-click your Form1.cs to bring up its designer. Now open the Toolbox on the right (or in View -> Toolbox). Select the "Viewer" component and drag it onto the Form Design area.

(If there is no Viewer in your Toolbox, please visit this page⁴ for troubleshooting.)

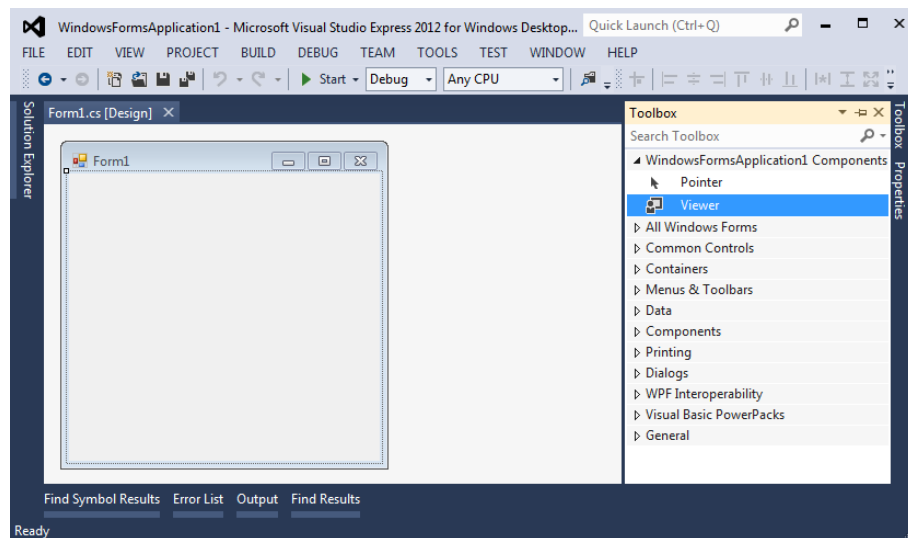


Figure 10: The Toolbox menu with the Viewer control.

3.2 Build and Run

Now it is time to test drive. Select "Start Debugging" from the "Debug" menu, or press F5. The application window should launch requesting a license key.

If you have already received a license key from us, you may now enter it. Otherwise, enter the key **novidhance**.

4. [http://vidview.imint.se/sdk/net/Toolbox Troubleshooting.html](http://vidview.imint.se/sdk/net/Toolbox%20Troubleshooting.html)

Please note that the license process requires a working connection with our servers. If that is a problem, see this page⁵ for information on how to issue a license for a computer that is not connected to the Internet.

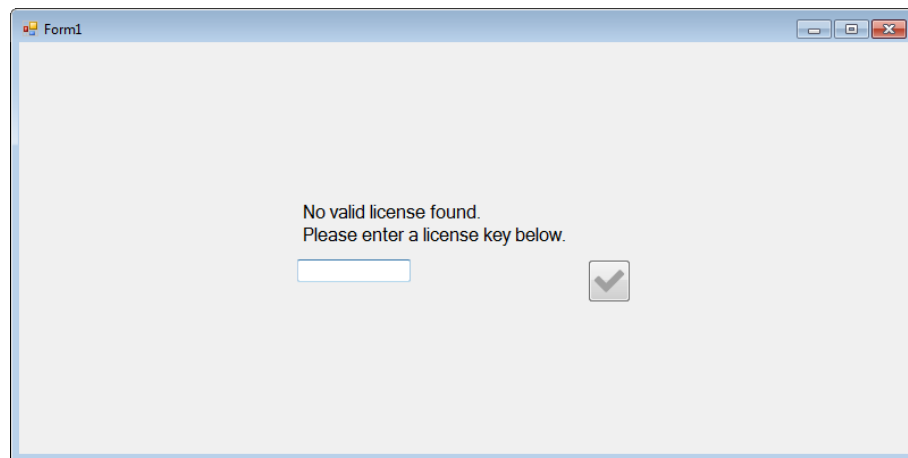


Figure 11: The enter a license key dialog.

3.3 Play a video file

Once the proper license is entered the open dialog is shown.

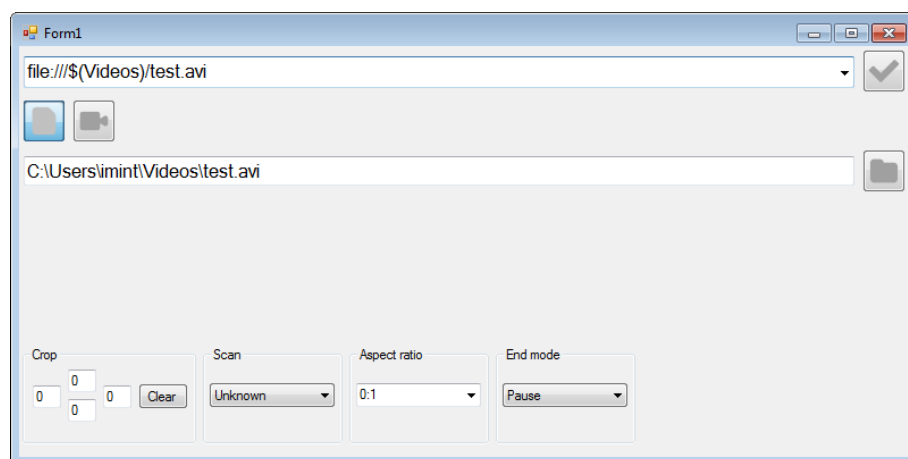


Figure 12: The open dialog.

5. <http://vidview.imint.se/license/Vidview%20License%20Management%20Guide.html>

In the application window, press the file folder icon. A standard file select dialog will open. Choose a video file on your computer, and click ok. Back in the application window, click the checkbox to the right of the top field. (Hint: if the application window is too small, re-size it in Visual Studio). This Vidview example will allow video files in an AVI-container with .avi file extension for mjpeg, wmv1, wmv2, mpeg4 v1 and dv, in ASF-container with .asf file extension for mjpeg, and in ASF-container with .wmv file extension in wmv1 and wmv2.

After selecting the file, the Vidview application window should look like this. Press the play button at the bottom to start video viewing.

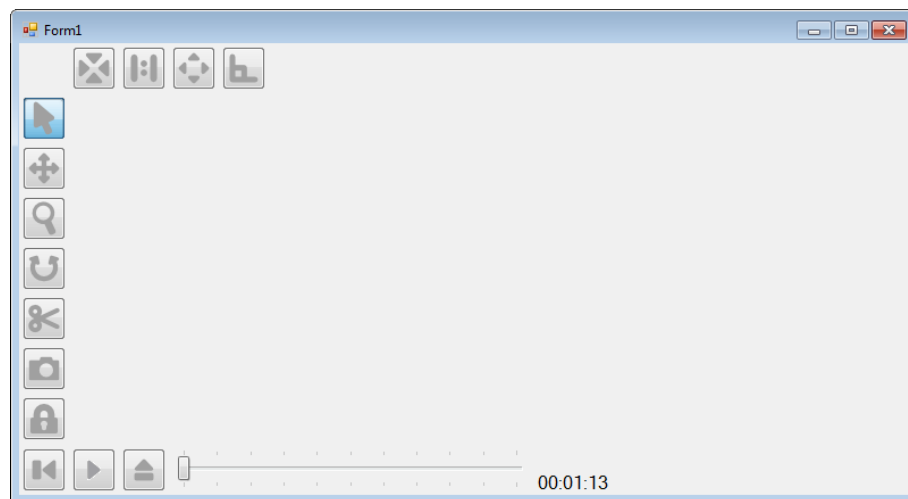


Figure 13: The application with opened video before pressing play.

You can now familiarize yourself with the basic video management functionality in the demo application, such as panning, zooming, rotating, replaying, etc.



Figure 14: Playback of rotated video.

4 Using Vidhance

4.1 Adding Vidhance Components

This section will describe how Vidhance video enhancement components can be added, and will use the Vidhance Scene Adaptive Contrast Optimization component as an example. The following step-by-step list will show how this can be added to the project created in the previous section.

4.1.1 Installing Vidhance packages

Return to the Manage NuGet Packages dialog by right clicking on the project in the Solution Explorer and selecting "Manage NuGet Packages...". Before installing any new packages, go to the **Update** tab on the left and make sure your packages are up to date. Then, go back to the **Online** tab, locate the **Imint.Vidhance.Contrast** package, and click "Install". It is found quickest by searching for **contrast** in the top right search field.

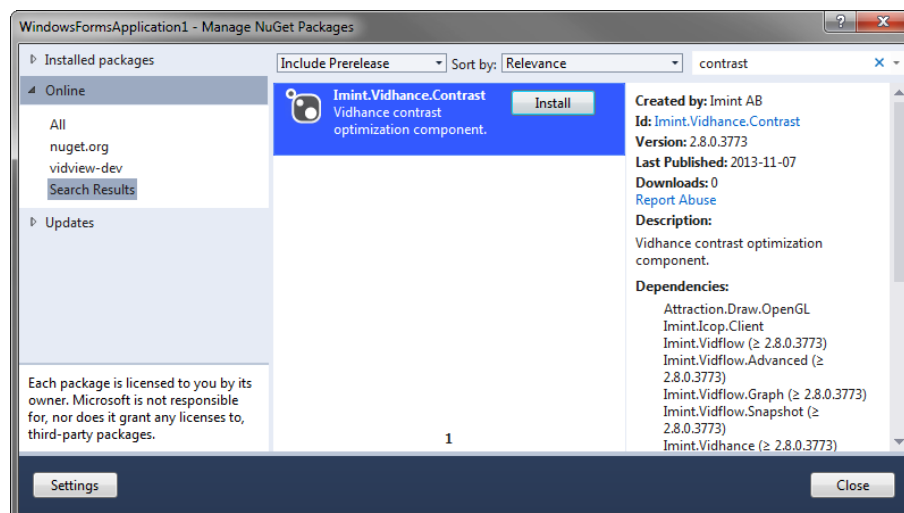


Figure 15: The Manage NuGet Packages dialog with the Imint.Vidhance.Contrast package selected.

Rebuild the project (F6) and start a debug session (F5) as described in the previous section.

4.1.2 Testing contrast optimizations

Follow the instructions in the previous section to select and open a video file.

In the Vidview demo application window, you will notice that a new icon has appeared to the left with a contrast setting symbol. Press the play button, and then the contrast button.

You can now experiment with the scene adaptive contrast functionality. You have user control buttons for disabling and enabling contrast enhancement, and to enable colorization. The left slider bar adjust contrast enhancement intensity, and the right slider bar the level of algorithmic complexity. High level of complexity typically pays off in very varying degrees of illumination in the scenery.

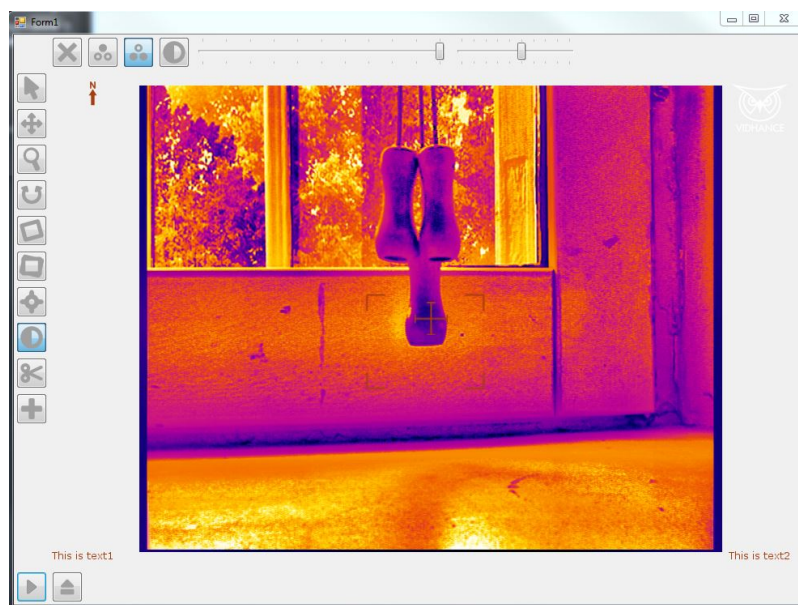


Figure 16: The Vidview sample application with regional contrast optimization and colorization enabled.

4.2 Process Separation

In order to protect the parent process Vidview can be run in a separate process. This way the operating system protects the parent process from crashing in the unlikely event of a Vidview failure.

1. Install the NuGet package `Imint.Vidview.Window`.
2. Open the `Viewer.cs` class and look for `this.vidview.SeparateProcess = false;`. Set this value to true.
3. Build and run.
4. In order to debug the new process, it must be manually attached to the debugger. Open **Debug -> Attach to Process** in the Visual Studio menu and select `Imint.Vidview.Window.exe` from the list of available processes.

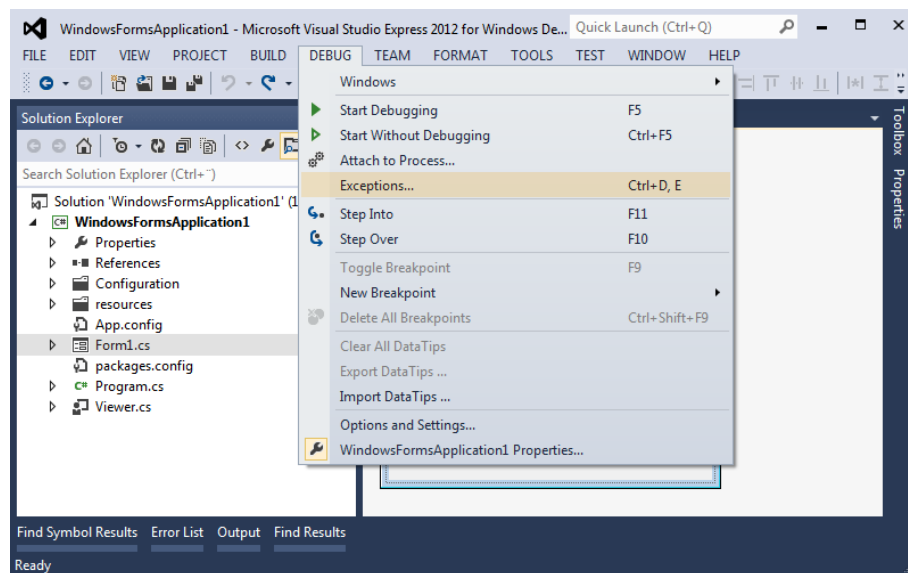


Figure 17: The Debug menu with the "Attach to Process..." option.

5 Further exploration

This Getting Started Guide will shortly leave you to your own devices for continued exploration. Here are a few suggestions for exploration:

5.1 Additional Vidhance Components

Examine other Vidhance video enhancement components, such as:

- stabilization
`Imint.Vidhance.Motion.Stabilize`
- local mosaicing
`Imint.Vidhance.Motion.Mosaic`
- deinterlacing
`Imint.Vidhance.Deinterlace`
- object tracking
`Imint.Vidhance.Motion.Track`

5.2 Experiment with video overlays

In the Solution Explorer, navigate to the folder `WindowsFormsApplication/Configuration/Overlay/Video`. Overlays are configured in XML files; explore the examples given.

5.3 GUI configuration

Look at GUI configuration options, also found under the `/Configuration` folder.

5.4 Live Video

If you want to experiment with live video, please contact Imint for further support for your specific set-up.

If you have a DirectShow device, such as a built-in web camera, you can use the basic Vidview Viewing application built here.

In the Open dialog of the Vidview application, enter the name of your device as: `directshow+capture:/// [name of device] /`, or select your device from the list.

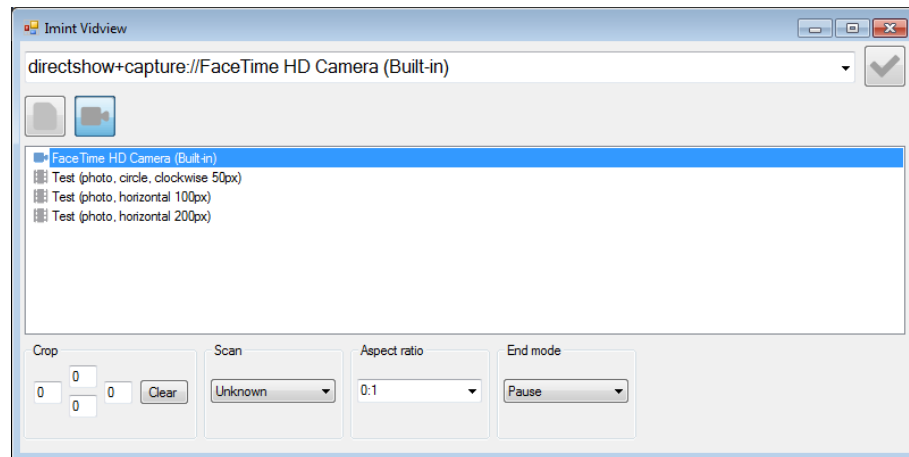


Figure 18: Example of how to open a web camera stream.

You should now be able to run your application with a live video stream.

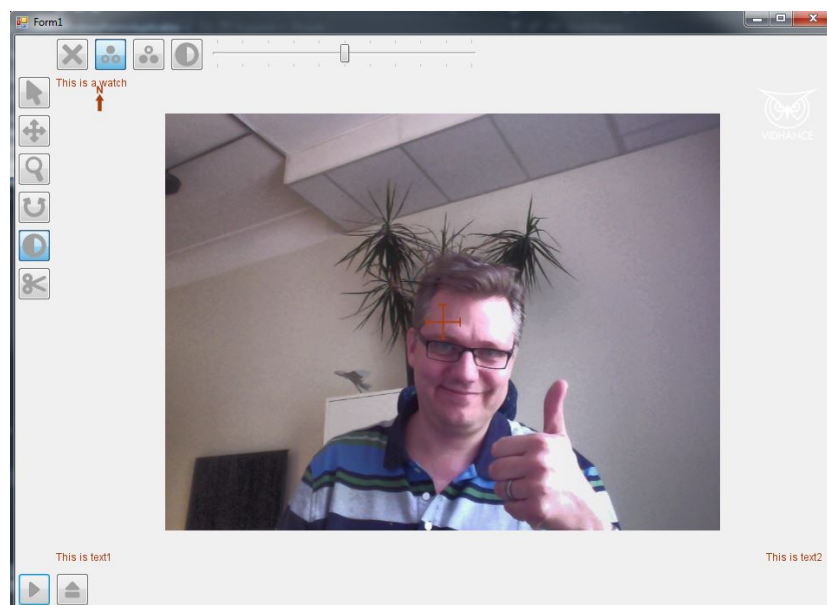


Figure 19: Vidview viewing video from built-in web camera.