Downloads you will need:
Visual Studio 2015, Community Edition
https://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx
(Make sure you include C++)

NDI SDK
http://pages.newtek.com/NDI-Developers.html

NDI Tools
http://pages.newtek.com/NDI-Tools-Pack-Download.html

NewTek Connect (for generating video from a webcam on another machine)
http://pages.newtek.com/NDI-Connect-Download.html

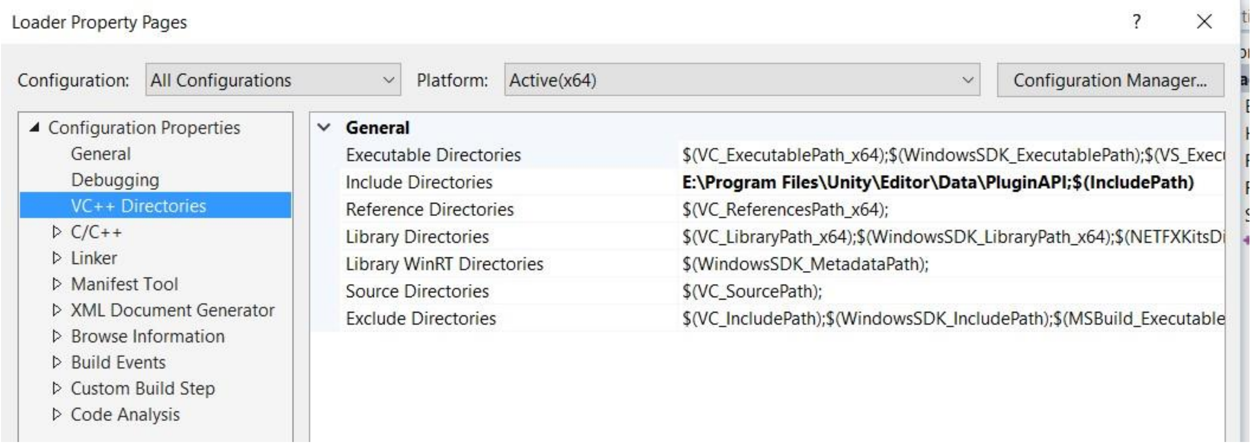
Unity, personal edition
https://unity3d.com/get-unity

Then, unzip the project files.
There should be two folders:

- Loader: This is the MSVC project for the loader dll that reads from NDI and writes to a texture.
- Example: This is the Unity project that invokes the loader.

Go to the Loader folder and double-click on Loader.sln to open it in MSVC 2015 (The free community edition.)

Configure the project to build by pointing it to the correct include folder for Unity (mine is on an E: drive. More typically it will be on C:)

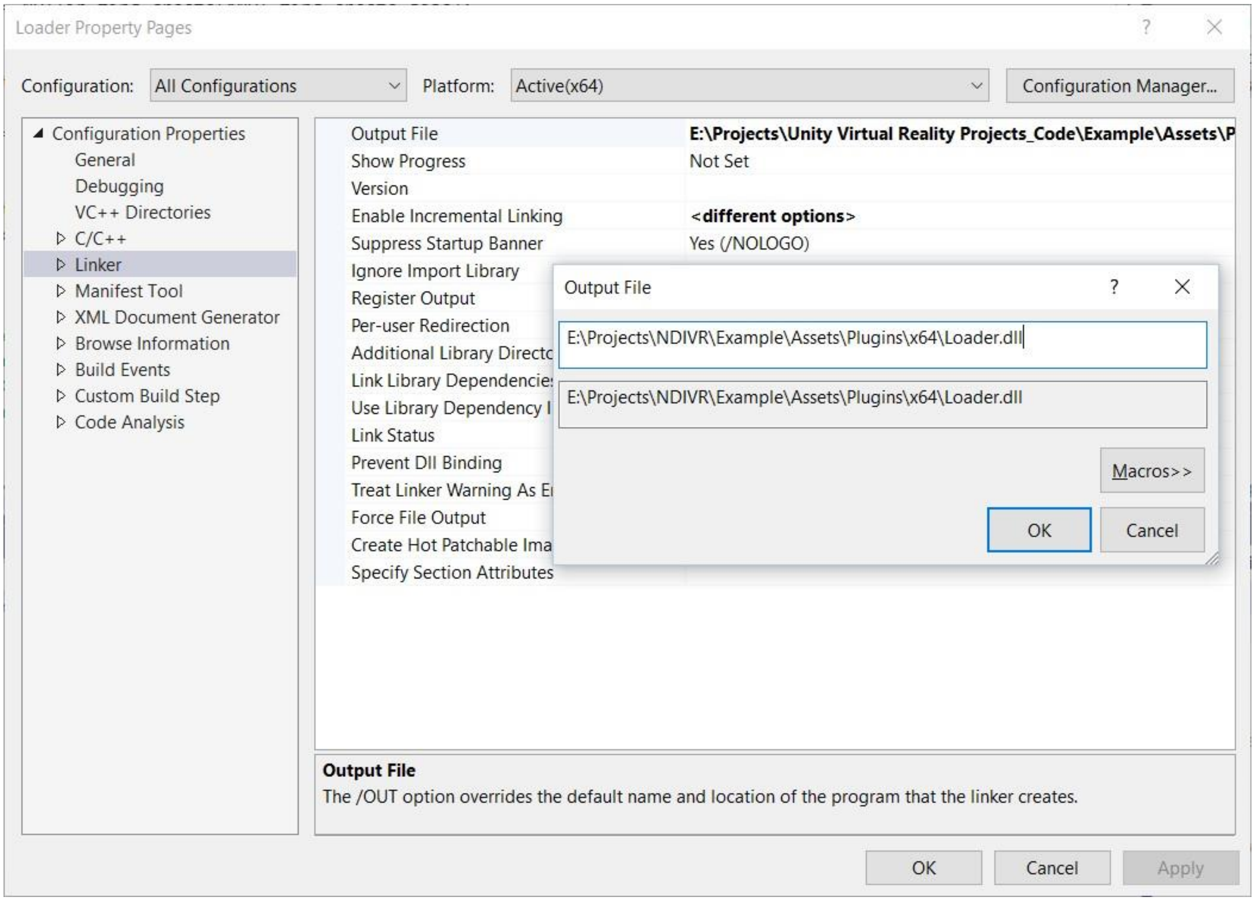


Loader Property Pages

Configuration: All Configurations   Platform: Active(x64)

General	
Executable Directories	$(VC_ExecutablePath_x64);$(WindowsSDK_ExecutablePath);$(VS_Exec
Include Directories	**E:\Program Files\Unity\Editor\Data\PluginAPI;$(IncludePath)**
Reference Directories	$(VC_ReferencesPath_x64);
Library Directories	$(VC_LibraryPath_x64);$(WindowsSDK_LibraryPath_x64);$(NETFXKitsDi
Library WinRT Directories	$(WindowsSDK_MetadataPath);
Source Directories	$(VC_SourcePath);
Exclude Directories	$(VC_IncludePath);$(WindowsSDK_IncludePath);$(MSBuild_Executable

And make sure that the output is being sent to the Unity Project's Assets location (more on that below – you can also view this location in the Unity project.)



Configuration: All Configurations
Platform: Active(x64)

Output File E:\Projects\Unity Virtual Reality Projects_Code\Example\Assets\P
Show Progress Not Set
Version
Enable Incremental Linking <different options>
Suppress Startup Banner Yes (/NOLOGO)
Ignore Import Library
Register Output
Per-user Redirection
Additional Library Directo
Link Library Dependencie
Use Library Dependency I
Link Status
Prevent Dll Binding
Treat Linker Warning As E
Force File Output
Create Hot Patchable Ima
Specify Section Attributes

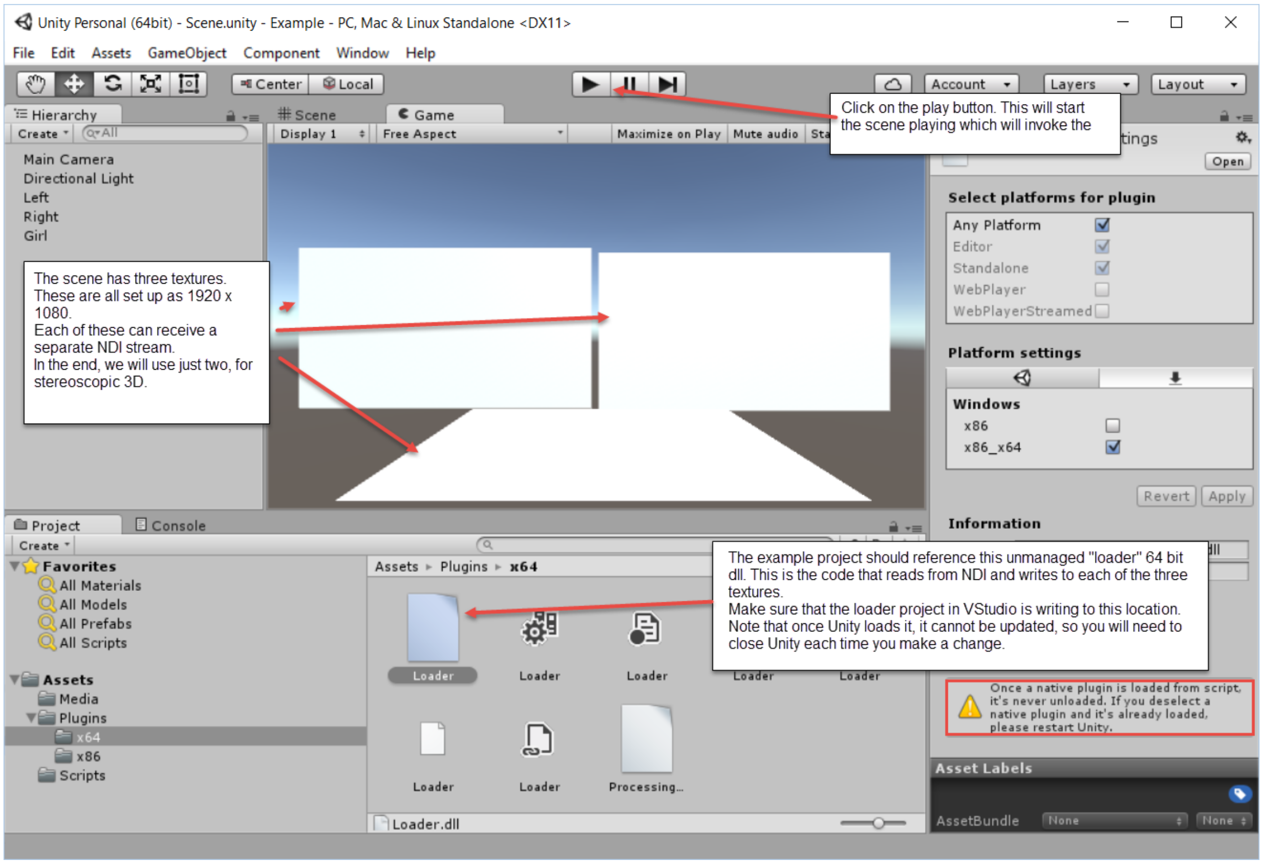
Output File

E:\Projects\NDIVR\Example\Assets\Plugins\x64\Loader.dll

E:\Projects\NDIVR\Example\Assets\Plugins\x64\Loader.dll

Output File
The /OUT option overrides the default name and location of the program that the linker creates.

Once this compiles, open Unity and chose the Open command to load from a file.
Select the Example folder. (Unity loads a folder as a project.)
Once it opens, click in the “Game” tab and it should look something like this:



Unity Personal (64bit) - Scene.unity - Example - PC, Mac & Linux Standalone <DX11>

Click on the play button. This will start the scene playing which will invoke the

The scene has three textures. These are all set up as 1920 x 1080.
Each of these can receive a separate NDI stream.
In the end, we will use just two, for stereoscopic 3D.

The example project should reference this unmanaged "loader" 64 bit dll. This is the code that reads from NDI and writes to each of the three textures.
Make sure that the loader project in VStudio is writing to this location.
Note that once Unity loads it, it cannot be updated, so you will need to close Unity each time you make a change.

Select platforms for plugin

- Any Platform ☑
- Editor ☑
- Standalone ☑
- WebPlayer □
- WebPlayerStreamed □

Platform settings

Windows

- x86 □
- x86_x64 ☑

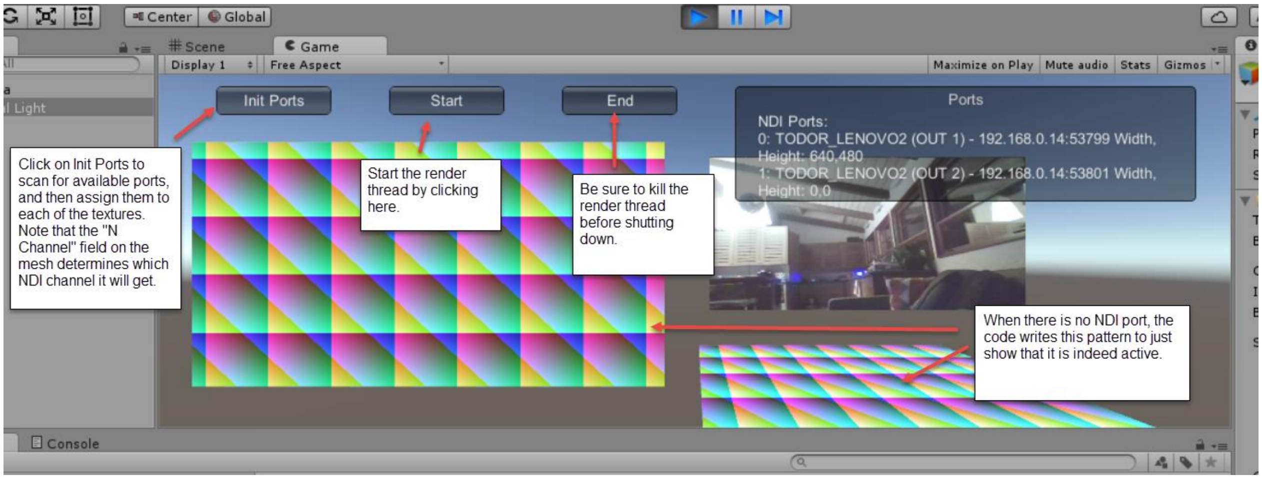
Once a native plugin is loaded from script, it's never unloaded. If you deselect a native plugin and it's already loaded, please restart Unity.

AssetBundle None None

This is a very simple project that has three textures which will receive video from NDI.

You are now ready to run it.

Click on the Play button in Unity.



Click on Init Ports to scan for available ports, and then assign them to each of the textures. Note that the "N Channel" field on the mesh determines which NDI channel it will get.

Start the render thread by clicking here.

Be sure to kill the render thread before shutting down.

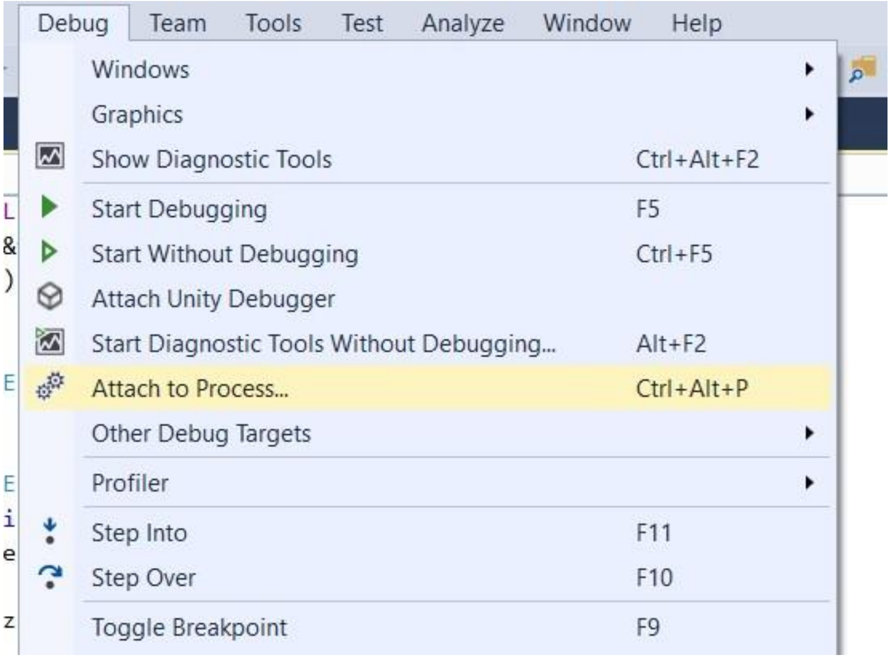
When there is no NDI port, the code writes this pattern to just show that it is indeed active.

```
Ports
NDI Ports:
0: TODOR_LENOVO2 (OUT 1) - 192.168.0.14:53799 Width,
Height: 640,480
1: TODOR_LENOVO2 (OUT 2) - 192.168.0.14:53801 Width,
Height: 0,0
```

To edit any of the scripts, stop playback and either double click on the script here in the assets, or invoke it by double clicking on “Example.sln” in the Example folder.

The three scripts are:
Loader – the wrapper functions for the dll calls.
Sample – code to manage each texture and set it up to read from NDI.
Toggle – the UI code, which handles the buttons and responds to each of them.

Debugging in the dll:
Make a Debug build. It should be 64 bit, not x86 (32 bit).
(Both the debug and release projects should output to the same file.)
From the dll Loader project (C++) choose Attach to Process. Be sure to use the regular attach, not the Unity debugger:



Set a break point where you want to step through the code.
Hit the run button in Unity.

The dll should load and hit your break point.