Then compile and link the generated code as a Python extension DLL. You will also need to link in the .lib for the wrapped DLL.

On Windows you do not link with a .dll file directly – you must use the accompanying .lib file instead. To do that go to Project -> Properties -> Configuration Properties -> Linker -> Additional Dependencies and add path to your .lib as a next line.

You also **must** make sure that the .dll file is either in the directory contained by the %PATH% environment variable or that its copy is in Output Directory (by default, this is Debug\Release under your project's folder).

If you don't have access to the .lib file, one alternative is to load the .dll manually during runtime using WINAPI functions such as [LoadLibrary](http://msdn.microsoft.com/en-us/library/windows/desktop/ms684175%28v=vs.85%29.aspx) and [GetProcAddress](http://msdn.microsoft.com/en-us/library/windows/desktop/ms683212%28v=vs.85%29.aspx).

**Umgebungsvariablen in Windows-Systemsteuerung einstellen**

PATH erweitern um Verzeichnis in dem sich die Vicon-dll befindet

**Vermeindlich fehlende Dependencies im dependency walker**

In addition to what @Ofek Shilon said, I usually ignore following dlls that dependency\_walker identified as missing when I try to find missing dlls for my program. You will see that your program runs fine when dependency\_walker says these dlls are missing.

* API-MS-WIN-\*.dll
* EXT-MS-WIN-\*.dll
* IESHIMS.dll
* EMCLIENT.dll
* DEVICELOCKHELPERS.dll
* EFSCORE.DLL
* WPAXHOLDER.DLL

java.lang.UnsatisfiedLinkError: V:\Hardware\Platformen\Mir\PROJECTS\ViconDataStream\libs\ViconDataStreamSDK\_CPP.dll: Can't find dependent libraries

If you load a 32 bit version of your dll with a 64 bit JRE you could have this issue. This was my case.

1 ) Go to ‘<http://tess4j.sourceforge.net/usage.html>’ click on -”Visual C++ Redistributable for VS2012 ” donwload it and run (VSU\_4\vcredist\_x64.exe or VSU\_4\vcredist\_x84.exe depending upon your system configuration )

2 put your dll files inside lib folder along with your libraries (ex \lib\win32-x86\your dll files).

-Djava.library.path="C:/MyLibPath;%PATH%"

%PATH% is your old -Djava.library.path