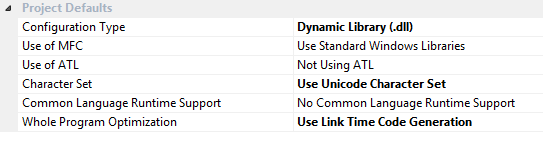
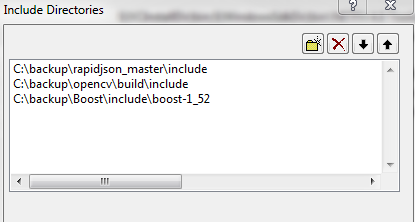
Based on easyPick, easyPickDLL is used to generate DLL, so that C# GUI application can call the algorithm easily. Here is how to setup the easyPickDLL project after creating the DLL project (Release mode).

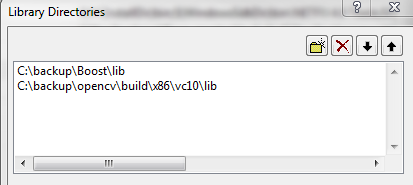
1. Make sure the opencv, Boost and rapidjson are installed and. Right click EasyPickDLL and select **Properties**. Make sure the project defaults are like this:



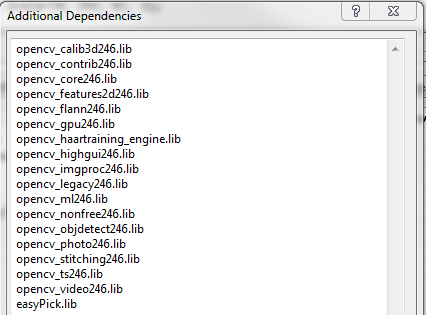
Then, select **VC++ Directories**. For **Include Directories**, add the include directories of opencv, boost, and rapidjson.



For **Library Directories**, add the .lib directories of opencv and boost.

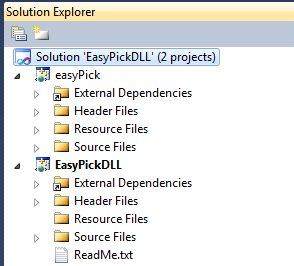


Back to Properties, go to **Linker -> Input -> Additional Dependencies**, add a list of lib files as demonstrated below: (also mentioned in the “how to setup opencv” documentation)



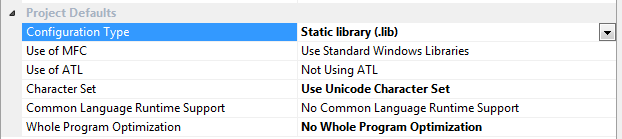
1. Include the easyPick project.

In the Solution Explorer, right click **Solution ‘EasyPickDLL’**, select **Add -> Existing Project**, then navigate to the easyPick project, click **Open**.



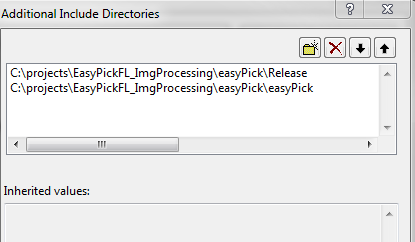
1. Build easyPick.lib if there’s no easyPick.lib file produced before.

Go to easyPick project, switch to “Release”. Go to **Property Pages -> Configuration Properties -> General**, change the Configuration Type to “Static library (.lib)”.



Build easyPick solution, in the Release folder, you will find  which will be used later to build DLL.

1. Go to **Properties -> C/C++ -> General -> Additional Include Directories**; add the directory of easyPick code files (.hpp and .cpp files). Also add the directory of easyPick.lib. In my local, for example:



Back to Properties, go to **Linker -> General -> Additional Library Directories**, and also add the directory of **easyPick.lib** like the example above.

1. Build solution, you should find in the Release folder.