1. Start Visual Studio 2010
2. File, New, Project from Existing Code...
3. Select "Visual C++" project type and click Next.
4. Enter project file location where the .cpp/.h/.i files are.
5. For Project Name, choose the name used in %module statement in your .i file (case matters).
6. Select project type "Dynamically linked library (DLL) project" and click Next.
7. Add to Include search paths the path to the Python.h file, usually something like "C:\Python27\include" and click Next.
8. Click Finish.
9. Right-click the Project in Solution Explorer, Add, Existing Item..., and select your .i file.
10. Right-click the .i file, Properties, and select Configuration "All Configurations".
11. Change Item Type to "Custom Build Tool" and click Apply.
12. Select "Custom Build Tool" in Properties (it will appear after Apply above).
13. Enter Command Line of "swig -c++ -python -outdir $(Outdir) %(Identity)" (this assumes SWIG is in your path and redirects the generated .py file to the Debug or Release directory as needed).
14. In Outputs enter "%(Filename)\_wrap.cxx;$(Outdir)%(Filename).py".
15. Click OK.
16. Right-click the .i file, and select Compile.
17. Right-click the project, Add, New Filter, name it "Generated Files".
18. Right-click "Generated Files", click Properties, and set "SCC Files" to "False" (if you use source-control, this prevents VS2010 trying to check in the generated files in this filter).
19. Right-click "Generated Files", Add, Exiting Item and select the \_wrap.cxx file that was generated by the compile.
20. Right-click the project, Properties.
21. Select Configuration "All Configurations".
22. Select Configuration Properties, Linker, General, Additional Library Directories and add the path to the python libraries, typically "C:\Python27\libs".
23. Select Configuration Properties, General and set TargetName to "\_$(ProjectName)".
24. Set Target Extension to ".pyd".
25. Build the "Release" version of the project. You can't build the Debug version unless you build a debug version of Python itself.
26. Open a console, go to the Release directory of the project, run python, import your module, and call a function!