1. These instructions assume OpenStudio is installed in “C:\Program Files (x86)\OpenStudio 0.9.6”.
2. Copy one or more of the C# examples from ‘C:\Program Files (x86)\OpenStudio 0.9.6\CSharp\examples’ to another location where you have read/write/execute file permissions. For this tutorial, we will work with the ‘Alpha1’ example project.
3. Open the ‘Alpha1.sln’ C# Solution file in the new location using Visual Studio 2008.
4. Ensure that the “Active solution platform” of the build configuration is set to “x86” and not “Any Computer” as the OpenStudio C# bindings are 32 bit only. The “Active solution configuration” may be either “Debug” or “Release”.
5. Select “Build->Build Solution”, the build process will fail with error messages related to missing OpenStudio references.
6. Use the Solution Explorer to navigate to “References->OpenStudio” notice that there is a warning icon on the OpenStudio reference. Right-click on the OpenStudio reference and select Remove. Now right-click on References and select “Add Reference”. Click on the Browse tab, then browse to and select “C:\Program Files (x86)\OpenStudio 0.9.6\CSharp\openstudio\OpenStudio.dll”
7. Select “Build->Build Solution”, the project should now build. However, when you attempt to run the program you will get an error such as:

"Unable to load DLL 'openstudio\_OpenStudioUtilitiesCore\_csharp': The specified module could not be found. (Exception from HRESULT: 0x8007007E)"

This is because the application is trying to load OpenStudio dlls that are not in your current path.

1. Programs on Windows search for dlls to load first in the same directory as the current application and then in the system paths. For now, copy all of the dlls in ‘C:\Program Files (x86)\OpenStudio 0.9.6\CSharp\openstudio’ to your build directory ‘\Alpha1\bin\Debug’ or ‘\Alpha1\bin\Release’. Now when you run the program everything should work.
2. When you distribute your program you will need to include the OpenStudio dlls as well as any Qt or other third-party dlls loaded by your application. A good tool to see which dlls your program uses is Dependency Walker, <http://www.dependencywalker.com/>. When you run this free tool, you can set the profile option to run your application with any arguments it takes and then see all of the dlls that were loaded as the program ran. You can then copy any dlls being loaded from your development environment into the distribution package for your end users.
3. Note that some OpenStudio components (notably the VersionTranslator and classes using the Qt SQLite plug-in such as the LocalBCL class) require additional runtime dependencies. We are working to reduce/simplify these dependencies to make it easier to deploy applications built on OpenStudio.