

Building Steem SSE on Windows

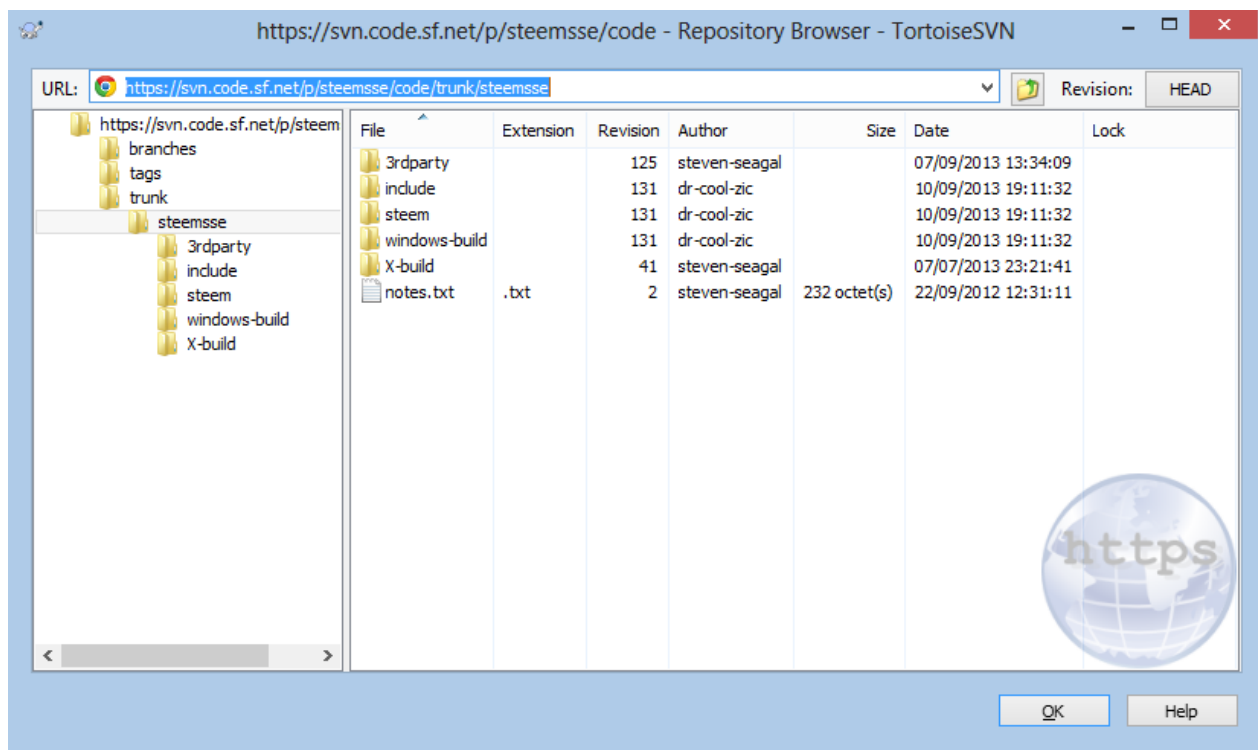
Tools Required

- I recommend that you use Visual Studio Ultimate 2012 – Version 11.0.60610.01 Update 3. Note that you can also use the free [Visual Studio Express 2012](#) from Microsoft.
- Steem is stored on Sourceforge using an subversion repository. To get the sources from Sourceforge I recommend that you use [TortoiseSVN](#).
- If you are using the **ultimate** version of VS2012 you can directly perform the SVN operation inside Visual Studio 2012 by using the free [VisualSVN](#) plugin (does not work on the express version of VS2012).
- The source tree already contains the required .obj files generated from Steem assembly sources. But in case you want to regenerate them you will need the [netwide assembler NASM](#)

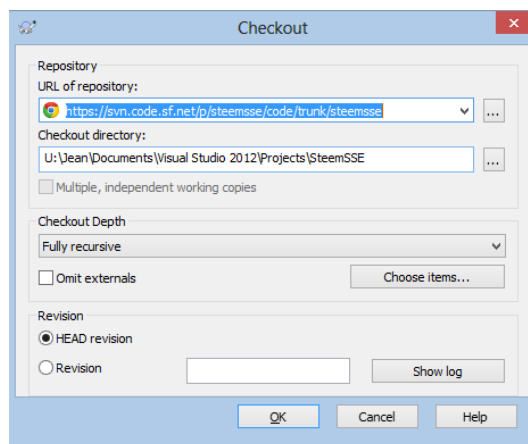
Steps to build Steem with the above tools.

Getting the sources

You first need to get the Steem sources from Sourceforge. The Steem project is located at <https://sourceforge.net/projects/steemsse/?source=directory>. You can browse the source tree with Tortoise svn browser either from <https://svn.code.sf.net/p/steemsse/code/trunk/steemsse> or from `svn://svn.code.sf.net/p/steemsse/code/trunk/steemsse`. For information the `svn+ssh://svn.code.sf.net/p/steemsse/code/trunk/steemsse` access does not seems to work (at least for me). Therefore if you are a developer I recommend that you use the https access.



To get the sources on your machine create a directory in your Visual Studio 2012 Project's repository. For example call it SteemSSE. Right click on the directory and select SVN Checkout command. Enter the correct URL for Steem repository and make sure the checkout directory is the one you have just created. Click OK and the checkout process should begin. At the end your SteemSSE directory contains all the sources and a green check mark indicates it is up to date. You are now ready to build Steem ☺



Building the .obj from the asm sources

This is a facultative step as the .obj are provided. But if you want to regenerate them follow these steps.

Assuming you have installed NASM at location C:\Program Files (x86)\nasm the only thing that you have to do is to go to the SteemSSE\steem\asm directory and double click on make_obj.bat file. This will create the asm_draw.obj and asm_osd.obj files in the SteemSSE\steem\obj directory.

Building Steem from the C++ sources

Now I assume that you have installed Visual Studio 2012 with C++ (using the Windows Desktop version).

Go to the SteemSSE\windows-build\VS2012 directory and double click the **SteemVS2012.sln** file. This will open the Steem project. You have the choice of building 4 different configurations:

- Release Configuration: This configuration builds an optimized version of Steem. This is the preferred configuration for standard user.
- Debug Configuration: This configuration build a non-optimized Steem with debug information to be used under the VS2012 debugger. This configuration is therefore useful for Steem developers.
- Boiler Configuration: This configuration builds an optimized version of Steem that also display the Steem internal debugger. This is useful for Atari program developers.
- Boiler Debug Configuration: This configuration build a non-optimized Steem with debug information to be used under the VS2012 debugger that also display the Steem internal debugger. This configuration is useful for developers

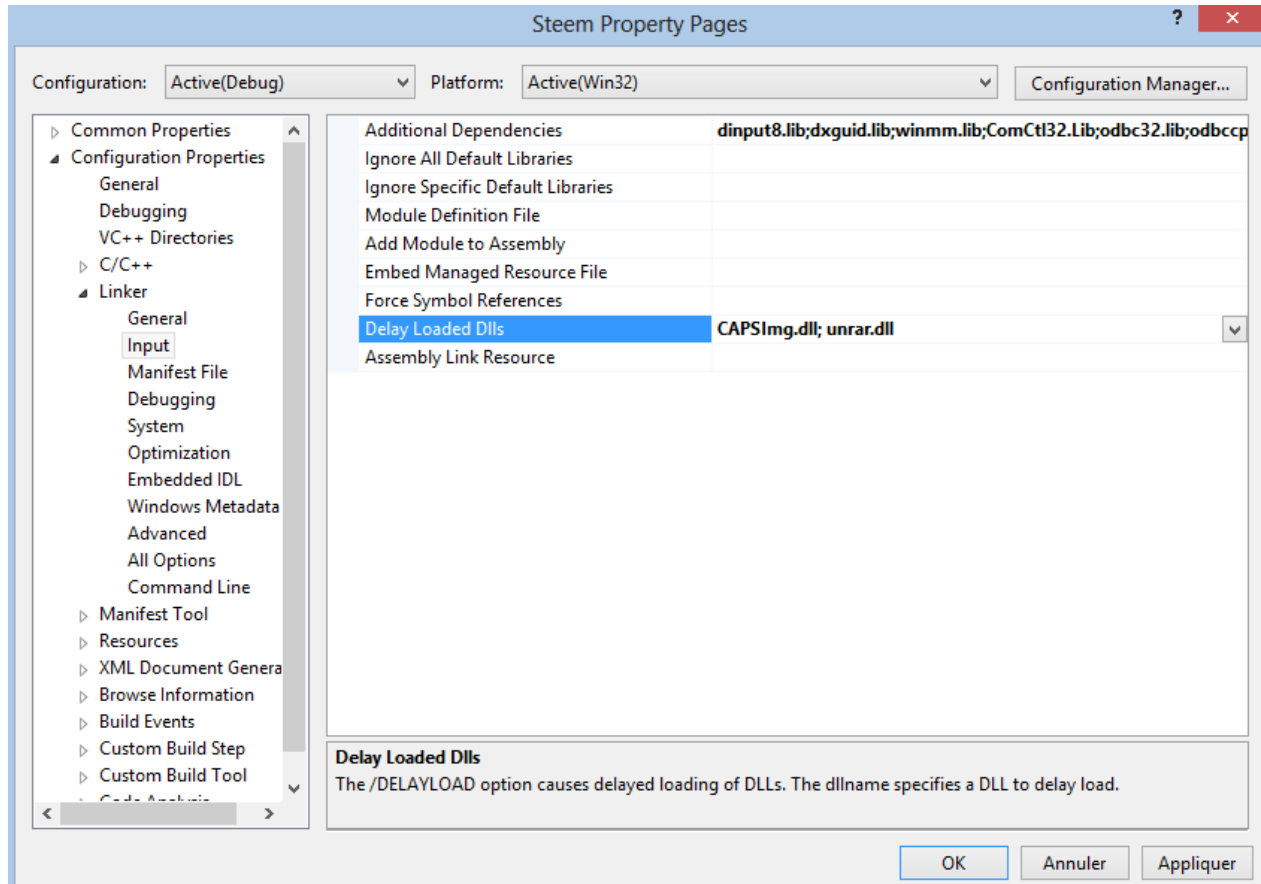
To build Steem select the desired configuration (i.e. Release) and build using for example the F6 key.

Visual Studio creates under SteemSSE\windows-build\VS2012 a directory corresponding to the selected configuration (Release, Debug, Boiler, BoilerDebug) that contains all intermediate files used by the compiler. It also create a directory called Executable that contains the Steem executable build according to the configuration: Steem.exe for Release, SteemDebug.exe for Debug, SteemBoiler.exe for Boiler, and SteemBoilerDebug.exe for BoilerDebug. For convenience all the required DLLs (CAPSImg.dll, pasti.dll, and unrar.dll) are also copied to the Executable directory.

Delayed DLL load

It is possible to use a Windows feature called delayed DLL to only load a DLL when you need it (for more information read ([Delay loading the DLL](#))). Personally I think this is a very bad idea. Steem needs several .dlls and any reasonable release of the Steem program should provide them. As already mentioned I have added a post process operation in VS2012 build that copy all the required dlls into the Executable directory so you have no excuse.

However if you like to live dangerously you can create a version with delayed loading DLL feature. For that you only need to change an option in the configuration file as shown below:



At the time of this writing only CAPSImg.dll and unrar.dll are required.

FYI What I have done to create the VS2012 release

- Created a Steem directory in my Visual Studio 2012 Project's repository.
- Checked out with TortoiseSVN the latest release of Steem in my Visual Studio 2012 Steem Project's repository.
- Created a VS2012 directory and copied VC6 required files then converted to Visual Studio 2012 format. In this directory is also located documentation and batch file used to build Steem with Visual Studio 2012.

- Added in steem/lib directory dinput8.lib and dxguid.lib files (old files no more released). Note the original Steem build was using dinput.lib but this lib is no more available in recent DirectX SDK. Using dinput8.lib instead seems to work fine.
- Added in steem/obj the files: asm_draw.obj, asm_osd.obj and make_obj.bat
- Modified several .cpp and .h file so the code compiled. All modifications are marked JLG VS2012 and should normally not break anything (mostly uninitialized variables and warning pragmas).
- Modified the configurations properties to compile correctly under VS 2012. The following configurations are available:
 - Boiler: Release version including Steem debugger
 - Boiler Debug: Debug version including Steem debugger
 - Debug: Debug version of Steem
 - Release: Release version of Steem (normally the version you want to use)

DrCoolZic – September 2013