**Compiling SSC+SSCdev for win32/x64/linux64/osx64**

**Using wxWidgets 3 (SVN), LK, and WEX**

**Aron Dobos – Updated 3 Jan 2012**

**This requires Visual Studio 2012 Express for Desktop:**

[**http://www.microsoft.com/visualstudio/eng/products/visual-studio-express-for-windows-desktop**](http://www.microsoft.com/visualstudio/eng/products/visual-studio-express-for-windows-desktop)

**wxWidgets 2.9 SVN, 32 bit.**

1. Create a folder c:\wxWidgets32
2. Right click on it, and check out (via TortoiseSVN) the wxWidgets trunk from the public SVN
   1. <http://svn.wxwidgets.org/svn/wx/wxWidgets/trunk>
   2. It’ll take some time to get all the files
   3. I checked out revision 73160. You can elect to use this revision from the SVN, or the head revision.
   4. When wxWidgets 2.9.5 comes out, or 2.9.6, or finally 3.0, we’ll migrate to using an official release instead of the SVN, but this is also good testing opportunity for the wxWidgets project ☺
3. Open Visual Studio 2012, and open the c:\wxWidgets\build\msw\wx\_vc9.sln file
   1. Allow the one-way upgrade, it will take some time to convert all the project files
   2. If the conversion report issues an error for the .sln, ignore it.
4. Open the folder C:\wxWidgets32\include\wx\msw in Windows Explorer
   1. Copy the setup0.h file to setup.h
5. Select the ‘Debug’ Configuration, and build.
6. Select the ‘Release’ Configuration, and build.
7. Set the WXMSW3 environment variable to c:\wxWidgets

Everything should complete successfully.

**wxWidgets 64 bit compilation:**

Repeat steps 1-7 above, with the following changes:

* Change the folder to “c:\wxWidgets64”,
* Open the project file wx\_vc9.sln, and let it convert. After it has converted, click on “Configuration Manager”, and under “Active Solution Platform” in the dialog box, click “new”. Select “x64”, and allow it to copy settings from the Win32 configuration.
* In Tools->Options->Projects and Solutions->Build and Run, change the maximum number of parallel project builds to “1”.
* Build both Debug and Release configurations for the x64 platform, all of them should complete successfully.
* Setup the environment variable “WXMSW3x64” to point to the c:\wxWidgets64

**LK Library**

1. Check out LK from <https://efmsvn.nrel.gov/lk/svn> into a folder called ‘lk’
2. Open the vc2012\_wx3\_unicode\lkwx3lib.sln project file
3. Build Release and Debug, both Win32 and x64.
4. Set the LKDIR environment variable to point to your LK folder.

**WEX Library**

1. Check out WEX from <https://efmsvn.nrel.gov/wex/svn/trunk> into a folder called ‘wex’
2. Open the wex\vc2012\_wx3\_unicode\wexlib.sln project
3. Build the Debug and Release configurations, both win32 and x64.
4. Set the WEXDIR environment variable to your wex folder.

Everything should complete successfully.

Now you should be able to compile the most recent SSC + SSCdev project for both 32 and 64 bit targets on Windows, using Visual Studio 2012.

**Mac OSX 10.8 notes:**

1. Download the latest XCode (4.6) from the App Store.
2. Run XCode
3. Select XCode->Preferences menu item
4. under “Downloads”, download and install all of the command line tools.
5. Check out wxWidgets
   1. mkdir wxWidgets
   2. svn co <http://svn.wxwidgets.org/svn/wx/wxWidgets/trunk> ./wxWidgets
6. Continue with LK and WEX

**Building an SSC SDK release package:**

To facilitate easy combining of binaries from all 3 target OSes for SSC, there is a subfolder in the SSC svn called sdk-release. This subfolder contains compiled binaries from all three platforms, and is updated when there is a new release of the SDK. The sdk-release folder is intended to be checked out on each target platform, the binaries compiled, copied over, and then checked in to the SVN. Then, when all the binaries are checked in, the folder can be zipped up for distribution containing all the different platform binaries.

1. In your c:\Projects\ssc folder, create a subfolder called sdk-release.
2. Right click on it, and check out <https://efmsvn.nrel.gov/ssc/svn/sdk-release>
3. On Windows:
   1. Compile SSC for both win32 and x64, Release configuration.
   2. Run the make-sdk-win.bat file, which will copy the binaries to the right places in the sdk-release folder.
4. On OSX: (10.8 64 bit)
   1. In the ssc/mac\_dylib folder, type make to build ssc64.dylib
   2. In the ssc/dev/osx\_wx3 folder, type make to build SSCdev.app
      1. This presumes you’ve compiled LK and WEX, as well as wxWidgets SVN with the following command line configuration:
      2. ./configure --prefix=/Users/adobos/local/wx3-svn --enable-stl=yes --enable-debug=no --enable-shared=no --with-cocoa --with-libjpeg=builtin --with-libpng=builtin --with-regex=builtin --with-libtiff=builtin --with-zlib=builtin --with-expat=builtin
      3. make
      4. make install
      5. sudo ln -s /Users/adobos/local/wx3-svn/lib/wx/config/osx\_cocoa-unicode-static-2.9 /usr/bin/wx-config-dsa
      6. sudo ln -s /Users/adobos/local/wx3-svn/lib/wx/config/osx\_cocoa-unicode-static-2.9 /usr/bin/wx-config-3
      7. cd lk/osx\_wx3\_unicode folder && make
      8. cd wex/osx\_wx3\_unicode folder && make
   3. Back in the toplevel SSC folder, type “bash make-sdk-osx.sh” to copy the right files to the sdk-release folder.
5. On Linux: (Assumed CentOS 6.3, 64 bit)
   1. In the ssc/ linux\_so folder, type make to build ssc64.so
   2. In the ssc/dev/linux\_wx3 folder, type make to build SSCdev.app
      1. This presumes you’ve compiled LK and WEX, as well as wxWidgets SVN with the following command line configuration:
      2. ./configure --prefix=/home/adobos/local/wx3-svn --enable-stl=yes --enable-debug=no --enable-shared=no --with-gtk=2 --with-libjpeg=builtin --with-libpng=builtin --with-regex=builtin --with-libtiff=builtin --with-zlib=builtin --with-expat=builtin
      3. make
      4. make install
      5. sudo ln -s /home/adobos/local/wx3-svn/lib/wx/config/gtk2-unicode-static-2.9 /usr/bin/wx-config-dsa
      6. sudo ln -s /home/adobos/local/wx3-svn/lib/wx/config/gtk2-unicode-static-2.9 /usr/bin/wx-config-3
      7. cd lk/linux\_wx3\_unicode folder && make
      8. cd wex/linux\_wx3\_unicode folder && make
   3. Back in the toplevel SSC folder, type “bash make-sdk-linux.sh” to copy the right files to the sdk-release folder and create the dependency and filetype information files.
6. Back on Windows,
   1. Update the sdk-release folder to get all the binaries for each of the platforms
   2. Run the make-sdk-common.bat file to copy the language wrappers, examples, common headers, and documentation files to the release folder
7. Zip up the sdk-release folder, and post it on the website! Avoid including any .svn subfolders in the zip file.

Hope that works!