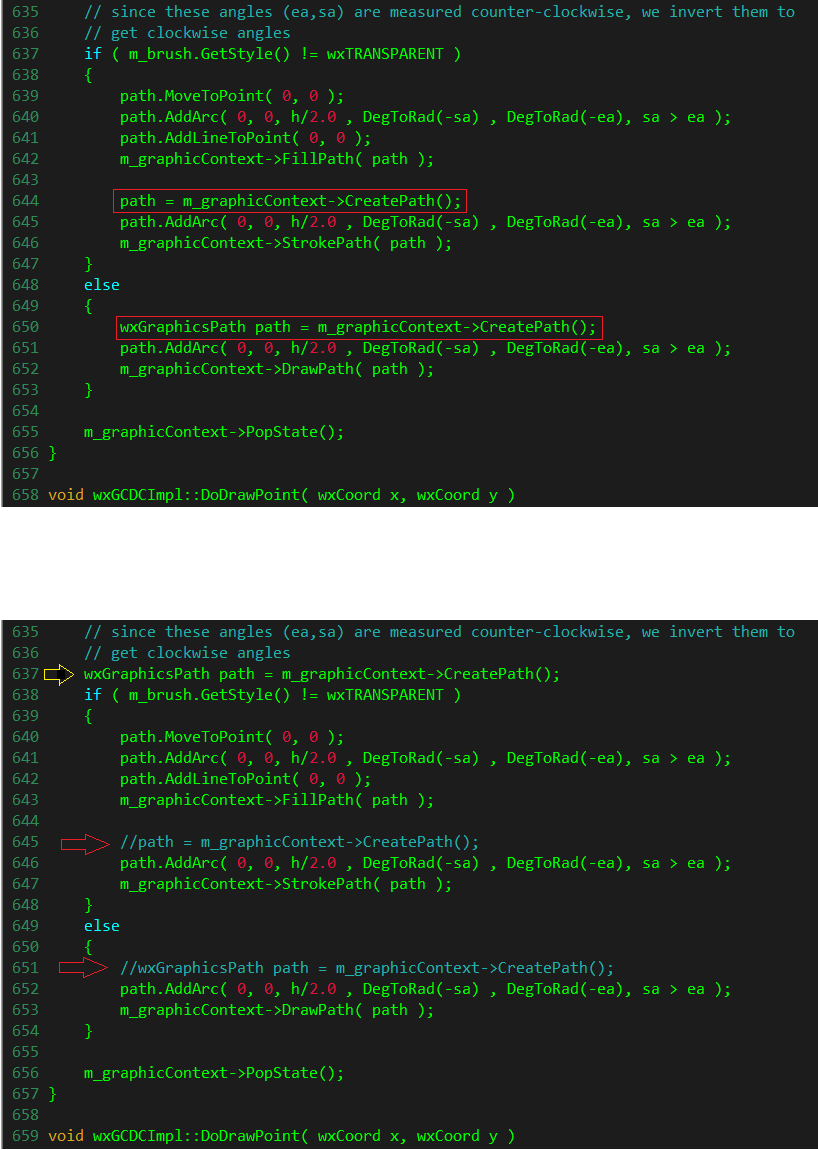
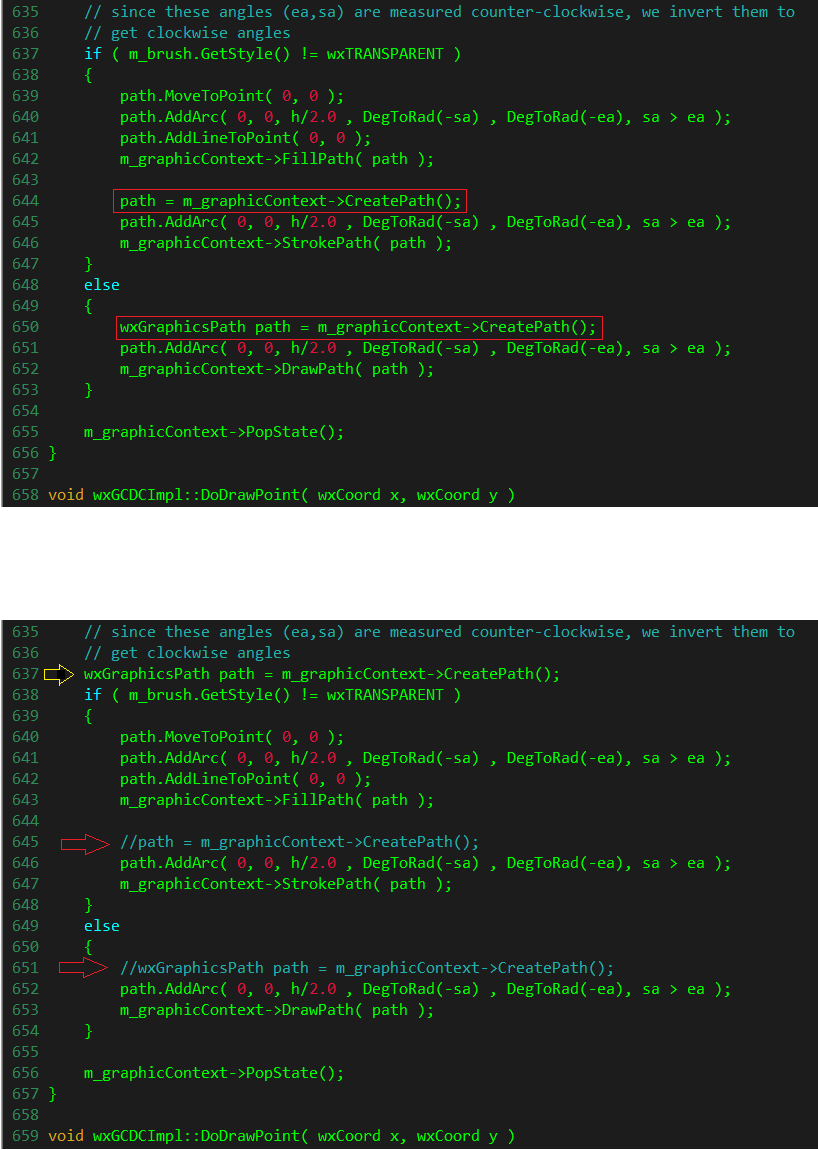
SAM Development with Visual Studio 2013 and wxWidgets 3.0.2

Aron P. Dobos, Severin Ryberg 20 February 2014

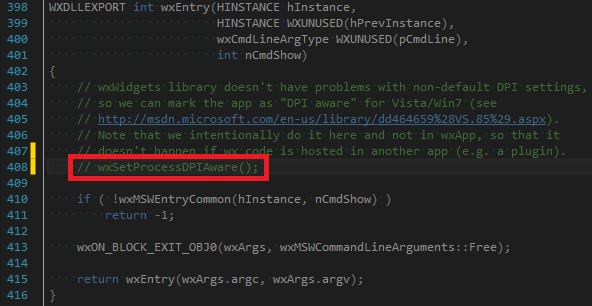
1. Download Visual Studio Express 2013 for Windows Desktop from <http://www.visualstudio.com/en-US/products/visual-studio-express-vs>. Usually it is more reliable to download the .iso disk image, and then use a utility like WinCDEmu to mount the .iso image to a virtual drive. You can download WinCDEmu from <http://wincdemu.sysprogs.org/>
2. Install VS 2013. Run VS 2013 and select Help->About->Product license information. Sign in or create a Microsoft.com account to obtain a free license.
3. Download wxWidgets 3.0.2 (wxWidgets-3.0.2.zip) from [http://sourceforge.net/projects/wxwindows/files/3.0.2/wxWidgets-3.0.2.zip/download](http://sourceforge.net/projects/wxwindows/files/3.0.1/wxWidgets-3.0.1.zip/download)
4. Extract to c:\wxWidgets-3.0.2
5. Implement a fix for the wxGraphicsPath issue. See related discussion at <http://trac.wxwidgets.org/changeset/78023>
   1. Open c:\wxWidgets-3.0.2\src\common\dcgraph.cpp
   2. Delete or comment lines 644 and 650 (Indicated by red boxes and arrows).



* 1. Insert the path declaration (which was originally on line 650) before the if statement on line 637 (Indicated by a yellow arrow).



1. Disable process DPI awareness for applications so that they are virtualized on high-DPI screens. See related discussion at <http://trac.wxwidgets.org/ticket/16116>
   1. Open C:\wxWidgets-3.0.2\src\msw\main.cpp
   2. Comment out line 408 that calls wxSetProcessDPIAware().



1. Open the wxWidgets solution file in Visual Studio and build
   1. Solution file location: C:\wxWidgets-3.0.2\build\msw\wx\_vc12.sln
   2. You will need to build Debug and Release configurations for both Win32 and x64 platforms. See Appendix C if you are unsure of how to build a VS solution.

Note: wxWidgets 3.0.2 does not require a separate folder for the 64-bit build anymore. The libraries for 32-bit go into lib\vc\_lib and 64-bit into lib\vc\_x64\_lib.

1. Setup an environment variable named WXMSW3 to point to c:\wxWidgets-3.0.2
   1. See Appendix A for clarification on setting up environment variables
   2. Also, delete your old WXMSW3x64 environment variable if you have it – it’s no longer needed.
2. Download the LK script engine from <https://efmsvn.nrel.gov/lk/svn/trunk>
   1. See appendix B for clarification on downloading files from external directories

Note: Suggested location: C:\Users\[User Name]\Documents\Projects\LK

Call this location [LK]

1. Build LK Engine
   1. Open solution file in VS: [LK]\ vc2013\_wx3\lkvc13wx3.sln
   2. Build the Debug/Release configurations for both win32 and x64.
      1. If successful, the libraries lkvc13wx3.lib, lkvc13wx3d.lib, lkvc13wx3x64.lib, and lkvc13wx3x64d.lib will be created in the “lk” folder.
2. Create a new environment variable called LKDIR that points to your [LK] folder.
   1. Follow instruction in appendix A
3. Download the wxWidgets Extensions (wex) library from <https://efmsvn.nrel.gov/wex/svn/trunk>.

Note: Suggested location: C:\Users\[User Name]\Documents\Projects\wex

Call this location [WEX]

1. Build the wex library
   1. Open solution file in VS: [WEX] \vc2013\_wx3\wexvc13wx3.sln
   2. Build Debug/Release configurations for both win32 and x64.
      1. If successful, the libraries wexvc13wx3.lib, wexvc13wx3d.lib, wexvc13wx3x64.lib, and wexvc13wx3x64d.lib will be created in the [WEX] folder, and dview{x64}.exe and sandbox{x64}.exe will exist in the solution folder, [WEX] \vc2013\_wx3.

NOTE: Both of the executables should run fine.

Also, it is a known issue when building this solution that the build will fail on a first attempt due to a missing library file, but will succeed if attempted again. This does not seem to be an issue.

1. Create a new environment variable called WEXDIR that points to your local [WEX] folder.
2. Download SAMnt files from <https://efmsvn.nrel.gov/SAMnt/svn/trunk>.
   1. Do not compile yet

Note: Suggested location: C:\Users\[User Name]\Documents\Projects\SAMnt

Call this location [SAMnt]

1. Create a new environment variable SAMNTDIR that points to the local [SAMnt] folder.
2. Download the SSC SDK from <https://efmsvn.nrel.gov/ssc/svn/trunk>.

Note: Suggested location: C:\Users\[User Name]\Documents\Projects\SAMssc

Call this location [SAMssc]

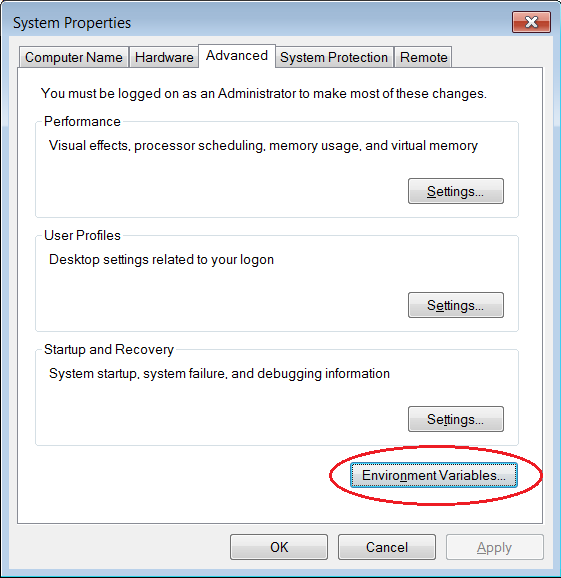
1. Build the SSC SDK
   1. Open solution file in VS: [SAMssc]\build\_vc2013\ssc\_vc2013.sln
   2. Build Debug/Release for both win32/x64.
      1. After compiling, the project file automatically copies the Release ssc.dll to the [SAMnt]\deploy\win32 or \*\x64 folder, so that SAM always is using the latest simulation engine library.
2. Create a new environment variable SSCDIR that points to the local [SAMssc] folder.
3. Build SAMnt
   1. Open solution file in VS: [SAMnt]\vc2013\_wx3\SAMnt\_vc2013.sln
   2. Build both Debug/Release and win32/x64.
      1. The sam.exe binaries should appear in [SAMnt]\deploy\win32 and \*\x64.
4. The SAMnt SVN contains precompiled binaries for libcurl which SAM uses to access the internet. Simply copy everything in [SAMnt]\vc2013\_wx3\libcurl\_ssl\_win32\bin to [SAMnt]\deploy\win32, and similarly for the x64 folders.

**Extra note:** The SAM registration process can be skipped by either:  
a) Entering “09332s” as the registration code in each version of SAM on your computer, or  
b) Run the command “regedit” from the command line, search for SAMnt, and add a new registry string value named “developer-registration” with the value 09332s, which will bypass the registration on every version of SAM installed on your computer.

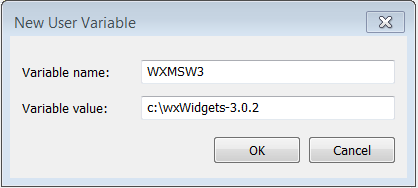
SAM should now run properly!

**Appendix A**: Setting environment variables

1. Open advanced system properties by following: start -> control panel -> system -> advanced system properties
2. Open the environment variable editor:



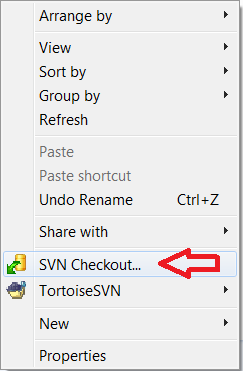
1. Click on ‘New…’
2. Set desired variable name (for example: WXMSW3) and value (for example: c:\wxWidgets-3.0.2). Be aware that the variable value field cannot accept spaces. Use an underscore instead.



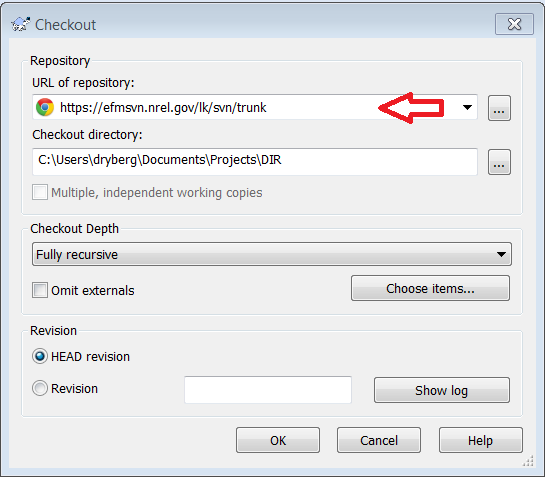
**Appendix B**: Downloading from external directories

There are many methods to download files from external directories. This method takes advantage of the ‘SVN Checkout’ option.

1. Create a directory where you wish to deposit the files from the external directory.
   1. Let’s call this location [DIR]
2. From within [DIR], right click and select ‘SVN Checkout…’

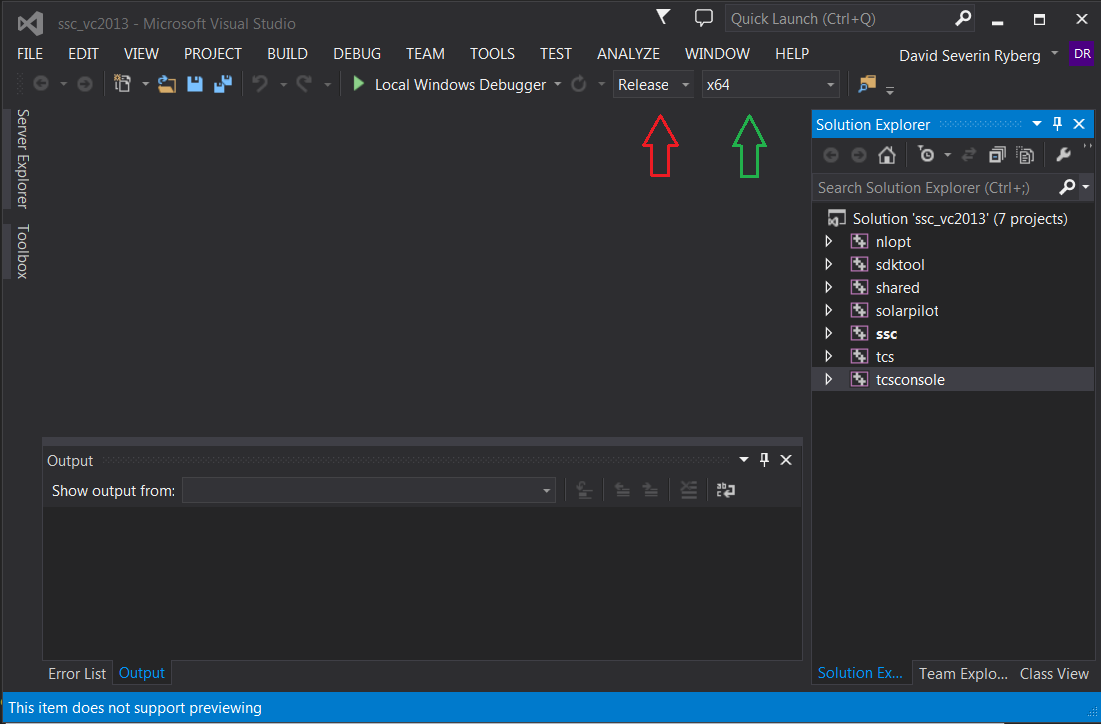


1. Paste the address to the external directory into the field labeled ‘URL of repository’. Then click ‘ok’.



**Appendix C**: Building Solutions in Visual Studio

1. After loading a solution file into Visual Studio, one can change the type of build by selecting options in the drop down menus indicated in the figure below.



* 1. The red arrow indicates the menu to choose a debug or release version
  2. The green arrow indicates the menu to choose win32 or x64

1. Build the solution by selecting BUILD on the top menu and then clicking on Build Solution.