**Build C++ Boost Libraries Set for Visual Studio 2013**

**Overview**

This project is intended for one to create an automate build environment to build the Boost libraries set to be used with VC++ in Visual Studio 2013.

I have created a set of make files to automate the build of the C++ Boost libraries using the `nmake` build tool that comes with Visual Studio. I have put the make files and other related files in github.

For this version, I’ve tested with `boost\_1\_58\_0` and Visual Studio 2013 community version. In a nutshell, the end product will be the Boost libraries set that built as the following flavors:

* static debug
* static release
* staticX64 debug
* static X64 release
* share debug
* share release
* shareX64 debug
* shareX64 release

The Boost libraries can either build as static or shared and Visual C++ projects can be configured as 32 bit or 64 bit. Also, the projects can be configured as debug or release. Since the Visual C++ linker is employing auto-link, depending on the current project configuration, it will look for a particular library name patterns to link with. The link will fail if the library with the particular name cannot be found.

As an example, to link with the `date time` static library in debug 32 mode, the linker will look for:

libboost\_date\_time-vc120-mt-sgd-1\_58.lib

Whereas for the shared 64 release mode, it will look for:

boost\_date\_time-vc120-mt-1\_58.lib

**Preparation**

* Download and install github for windows. The URL:
  + <https://windows.github.com/>
* Download and install Visual Studio community version. The URL:
  + <https://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx>
* Download and install the Microsoft Message Passing Interface (MS-MPI). The URL:
  + <https://msdn.microsoft.com/en-us/library/bb524831%28v=vs.85%29.aspx>
* Download and install python 2.7.x *for windows* (I am using version 2.7.9). The URL:
  + <https://www.python.org/downloads/release/python-2710/>
  + **NOTE**: Ensure that python folder is added to the %PATH% environment variable.
* Download and save the `boost\_1\_58\_0` bundle (Make sure to download this ZIP file: “**boost\_1\_58\_0.zip**” – The name is very important). The URL:
  + <https://sourceforge.net/projects/boost/files/boost/1.58.0/>

**Procedure**:

**Setup gihub and clone the repository**

* You need to create an account in github and sign in using the `GitHub` program (If you not have an account in github yet, create one from <https://github.com/>).
* Start the `git shell` program and navigate to the folder where the Boost libraries are going to be built. Type this command to clone the repository locally:

git clone https://github.com/stevenong2006/BoostBuild\_VisualStudio2013

* As the result, the following folder/file hierarchy is created:
  + BoostBuild\_VisualStudio2013 (folder)
    - src (folder)
      * 7z.dll
      * 7z.exe
      * bzip2-1.0.6.tar.gz
      * Expat2.1.0.zip
      * icu.7z
      * user-config.jam
      * zlib-1.2.8.tar.gz
    - Config.mnk
    - Makefile\_BoostLib.nmk
    - Makefile\_main.nmk
* Copy/move the “**boost\_1\_58\_0.zip**” file into the `src` folder

**Build the Boost libraries set using nmake tool**

* From the installed Visual Studio, start “Developer Command Prompt for VS2013”
* Navigate to the newly created folder `BoostBuild\_VisualStudio2013` where the ` Makefile\_main.nmk` is. Type the following command:

nmake /f Makefile\_main.nmk all

* This will kick-off the build process. All the zip files will be extracted and the boost libraries set will start to build. The build activities will be logged into the `Boost\_Build.log` for the record. If you have `tail` installed, you can view the activities at real time by:

tail –f Boost\_Build.log

* As the result, a folder called `boost\_1\_58\_0\_MsVCBuild` is created. If the build goes successfully, you will find the mentioned Boost libraries set in this folder.

To start, you can look at the available libraries at (<http://www.boost.org/doc/libs/>) and learn how to use them.

Enjoy,

Steven