

# Compiling mauveAligner on Windows

This document describes how to configure a Windows computer to build the command-line mauveAligner and related binary programs. Several software development tools must be installed, all of which are available free of charge as of 11/11/2006. Building parallel OpenMP and 64-bit versions of the code requires a non-free version of Visual Studio Professional. There are two primary ways to build the software, either through the Visual Studio GUI or from a command-line script.

## Step 1. Install Visual Studio C++

Download and install the free visual studio c++ express edition and platform SDK step by step instructions:

<http://msdn.microsoft.com/vstudio/express/visualc/usingpsdk/>

Installation must be done as a user with admin rights. Viz studio must be run at least once by an admin after installation to configure paths and registry values.

Alternatively, install a commercial licensed copy of visual studio to do 64-bit and OpenMP builds.

## Step 2. Install a subversion client

In this example we will use TortoiseSVN:

<http://tortoisesvn.tigris.org/>

TortoiseSVN requires admin privs to install

Using the command-line build script requires command-line subversion (the apache flavor), available from:

<http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=8100>

## Step 3. Install the Boost C++ development libraries

The latest version of boost can be installed with the installers from <http://boost-consulting.com> . At present, the latest version is 1.35.0, available here:

[http://www.boostpro.com/boost\\_1\\_35\\_0\\_setup.exe](http://www.boostpro.com/boost_1_35_0_setup.exe)

Alternatively, boost can be built from source. This is necessary for 64-bit windows boost builds if the installer does not include them. The following instructions are for the 1.35.0 release. Download boost\_1\_35\_0.zip and boost-jam (boost-jam-3.1.13-1-ntx86.zip) from

[http://sourceforge.net/project/showfiles.php?group\\_id=7586](http://sourceforge.net/project/showfiles.php?group_id=7586)

Extract boost to a new development folder (C:Developmentboost). Extract the bjam binary from the boost-jam zip file to the newly created C:Developmentboost directory. Make the changes to vsvars32.bat that are given at the bottom of [http://boost.org/tools/build/v1/vc-8\\_0-tools.html](http://boost.org/tools/build/v1/vc-8_0-tools.html). Admin privileges are necessary to edit vsvars32.bat, and using "Run As..." on your favorite text editor can be helpful here. Next, open a command prompt (Start Menu->Run "cmd.exe"), and switch to the boost folder "cd C:Developmentboostboost\_1\_35\_0". Start the boost build with the command:

```
bjam --prefix=C:Developmentboostboost_1_35_0" --toolset=msvc release address-model=32 runtime-link=static link=static  
threading=multi --with-filesystem --with-iostreams --with-program_options --with-system -a install
```

```
bjam --prefix=C:Developmentboostboost_1_35_0" --libdir=C:Developmentboostboost_1_35_0lib64" --toolset=msvc release  
address-model=64 runtime-link=static link=static threading=multi --with-filesystem --with-iostreams --with-program_options  
--with-system -a install
```

The build may take hours to complete on a slow machine. Once completed, the libraries will reside in C:Developmentboost and the build directories can be deleted (boost\_1\_35\_0stage and boost\_1\_35\_0bin) to save disk space.

## Step 4 (GUI only). Check out Mauve-related source code repositories

If you are only interested in the automated build script, skip to Step 4 below. Otherwise, create a development folder e.g. "C:Development" and right-click inside the development folder. Select the "SVN checkout" menu item. A dialog box will pop up asking the location of the repository to check out, and the destination directory. Perform a checkout for each of the following

repositories:

## Repository    Checkout Directory

<https://mauve.svn.sourceforge.net/viewvc/mauve/>

## Step 5 (GUI only). Configure Visual Studio paths

In order to build mauve-related software, it is necessary to set the include header and library search paths. To do so, launch visual studio and from the menu bar, select "Tools->Options..." In the dialog that appears, select the "Projects and Solutions"->"VC++ Directories" panel. Then select "Show directories for:"->"Include files" in the preference panel. Now click the new folder icon and add the following directories:

- \* C:DevelopmentlibGenome
- \* C:DevelopmentlibClustalW
- \* C:Developmentmuscle
- \* C:DevelopmentmusclelibMUSCLE
- \* C:DevelopmentlibMems
- \* C:Developmentboostboost\_1\_35\_0

If source code was installed in a directory other than C:Development, change the paths accordingly. Then select directories for library files ("Show directories for:"->"Library files") and add the following paths:

- \* C:DevelopmentlibGenomelib
- \* C:DevelopmentlibClustalWlib
- \* C:Developmentmusclelib
- \* C:DevelopmentlibMemslib
- \* C:Developmentboostboost\_1\_35\_0lib

On 64-bit windows, replace the boost library path with the path to 64-bit libraries, e.g. "C:Developmentboostboost\_1\_35\_0lib64"

## Step 6 (GUI only). Build mauveAligner, progressiveMauve, procrastAligner, muscle etc.

Open the file mauveAligner/projects/everything.sln with visual studio. The workspace contains an assortment of projects related to mauve. To build all projects, right-click on the text "Solution 'everything'" in the project listing at left and select "Build Solution" from the popup menu. A batch build can be used to build for several targets at once (menu "Build"->"Batch Build"). When the build succeeds, binaries will generally be in the mauveAligner/projects or mauveAligner/bin directories.

## Step 4 (Script only). Configure and run the build script

Download the latest windows build batch script from:

[http://mauve.svn.sourceforge.net/svnroot/mauve/build\\_scripts/build\\_win.bat](http://mauve.svn.sourceforge.net/svnroot/mauve/build_scripts/build_win.bat)

Save the batch script to a temporary build directory, for example C:build\_temp. Then right click on the build script to edit it. You must now configure the program paths near the top of the build script. Once you have finished doing so, simply run the script from a windows command-prompt: build\_win.bat