# Steps and info for library installs and builds.

A public repository for files/code related to the project was setup at:

<https://code.google.com/p/roadrunnerwork/>

The repository contains code and libraries relevant to this project, as well as svn externals to required libraries, i.e. libsbml, NOMLib, SBW Core and libstruct. In the following, when referring to a checked out working copy, its root folder is denoted: svn\_root.

Default output of libraries and DLL's are set in CMake to be c:\sbw\_libs, i.e. the CMAKE\_INSTALL\_PREFIX flag.

## 1. Compiling libsbml

The libsbml code is located at svn\_root/Externals/libsmbl and the actual address to the libsbml external is https://sbml.svn.sourceforge.net/svnroot/sbml/trunk/libsbml.

Short instructions on how to compile libsmbl is in the document: svn\_root/Docs/libSBMLBuildWindows.docx   
which is an excerpt from the main libsbml documents. One important step mentioned in that document is to download win-dependencies. In the following only win32 (not x64) is being considered.

The CMake GUI is used to generate solution files for Visual Studio.

When running CMake, the following settings were used:  
Generator for project: VisualStudio10 and option Use default native compilers

The following environment flags in CMAKE were modified:

CMAKE\_INSTALL\_LIBDIR=C:/sbw\_libs/libsbml/lib  
CMAKE\_INSTALL\_PREFIX = C:/sbw\_libs/libsbml

Clicking generate in CMake creates VisualStudio project files in the folder svn\_root/Externals/libsbml/build.

The solution file **libsbml.sln** is opened in Visual Studio and the target ALL\_BUILD project can be built. After finishing successfully, the INSTALL target is built, which copies all relevant libs, and dlls using the folders defined above by CMAKE\_INSTALL\_LIBDIR and CMAKE\_INSTALL\_PREFIX, i.e. C:/sbw\_libs/libsbml.

## 2. Compiling NOMLib

NOMLib is part of Snowburst, and is *a C API to libSBML that implements the original NOM API used in SBW*, from documentation at <http://snowburst.googlecode.com/svn/NOMLib/help/html/index.html>

Snowburst is found in the externals folder, i.e. svn\_root/Externals/snowburst. Actual repository address is: https://snowburst.googlecode.com/svn.

NOMLib is built using Visual Studio by the solution file: NOMLib.sln, found in ./snowburst/NOMLib

After compiling, NOMLib binaries and link libraries are found in:  
svn\_root\Externals\snowburst\NOMLib\bin\Debug

## 3. Compiling SBW core

The SBW core is found in svn\_root/Externals/sbw\_core with actual repository address: http://sbw.svn.sourceforge.net/svnroot/sbw/trunk/core

To build, open the solution file SBW-vs2010.sln found in folder: svn\_root\Externals\sbw\_core\VisualStudio

This build creates a SBWD.dll in .\sbw\_core\bin and import library SBWD.lib in sbw\_core\lib, needed for building libstruct, see below.

## 3. Compiling libStruct

libStruct is a C/C++ library. It is found in svn\_root/Externals/libStruct with actual address https://libstruct.svn.sourceforge.net/svnroot/libstruct/

In the following, instructions found in the Readme.pdf file in the root folder of libStruct was consulted. Main strategy is to use CMake to generate project and solution files for Visual Studio.

Running CMake GUI and pointing it to libStructs root folder, and creating a folder, build, in the same for the CMake output files, a few errors occurs, which need attention:

CMake will complain that the following flags are undefined:

CLAPACK\_F2C\_LIBRARY  
 LIBSBML\_INCLUDE  
 LIBSBML\_LIBRARY  
 SBW\_INCLUDE  
 SBW\_LIBRARY

To fix the above;  
 CLAPACK\_F2C\_LIBRARY is pointed to ./libstruct/dependencies/lib/clapack.lib  
LIBSBML\_INCLUDE is set to SBML includes, i.e. C:/sbw\_libs/libsbml/include  
LIBSBML\_LIBRARY is set to C:/sbw\_libs/libsbml/lib/libsbml.lib  
SBW\_INCLUDE is set to svn\_root\Externals\sbw\_core\include  
SBW\_LIBRARY is set to svn\_root\Externals\sbw\_core\lib\SBWD.lib

The visual studio solution file, libstruct/build/LibStructural.sln is opened in VisualStudio and target ALL\_BUILD can be built. If finished, successfully, the INSTALL target is executed, which installs the dll's and libraries to the folder specified in CMakes CMAKE\_INSTALL\_PREFIX.