Installing / Building LibSBML

Frank T. Bergmann

Sarah Keating

http://sf.net/projects/sbml/files/libsbml/5.0.0

Binary Installers

INSTALLING LIBSBML

Binary Installers

Windows

libSBML-5.0.0-win-x64.exe
libSBML Setup
SBML Team

libSBML-5.0.0-win-x86.exe
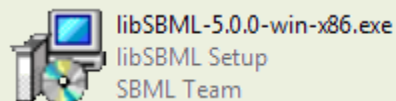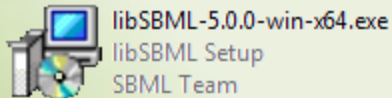libSBML Setup
SBML Team

OS X

libSBML–5.0.0–libxml2–
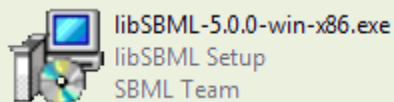snowleopard.pkg

DEB

rpm

Linux

Windows Installers
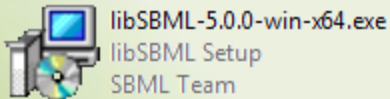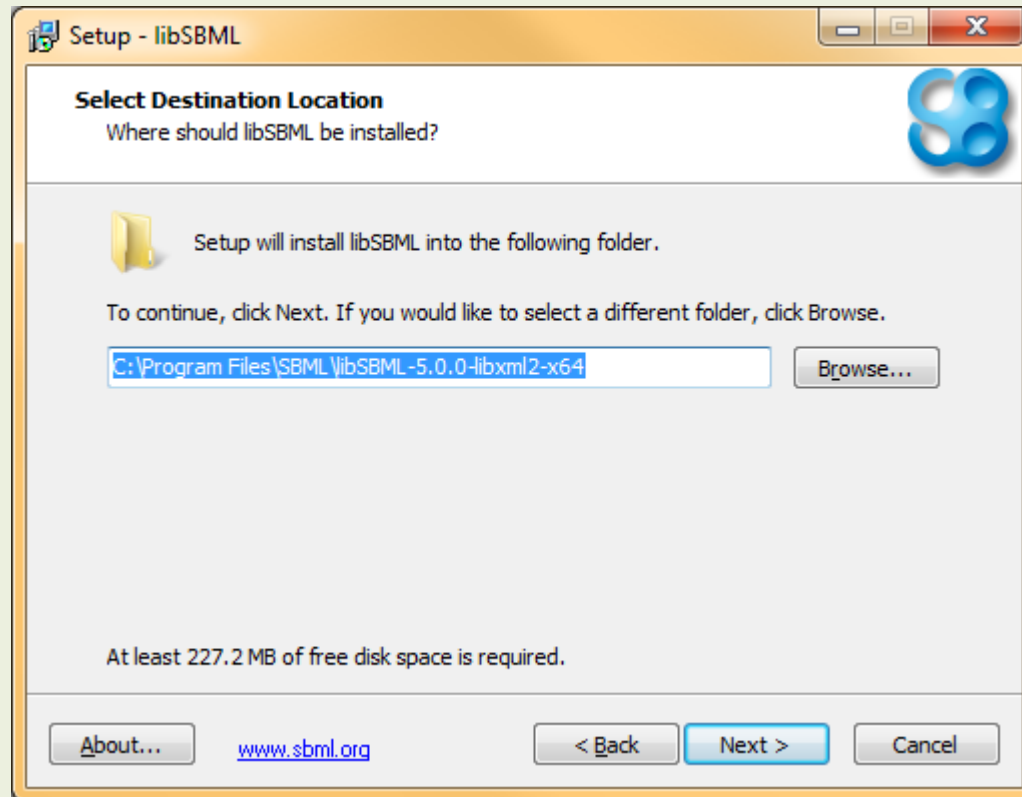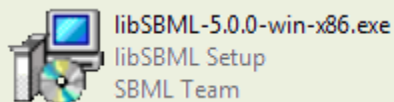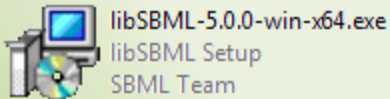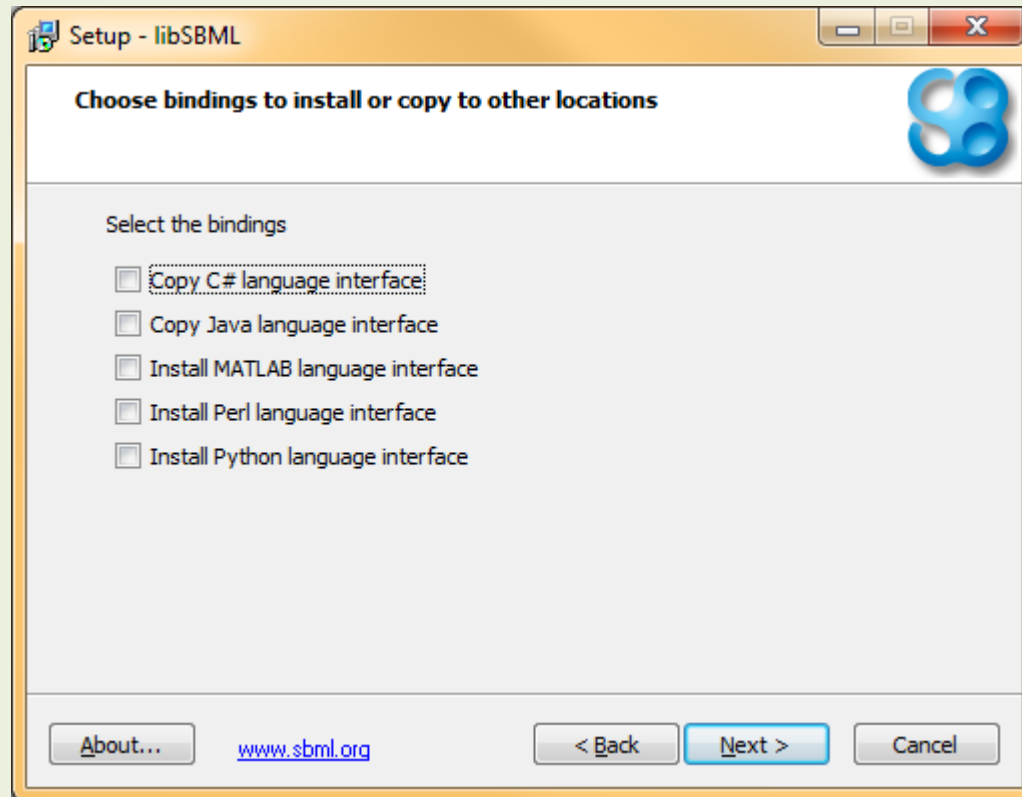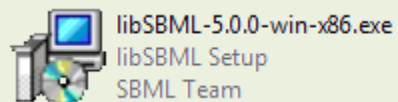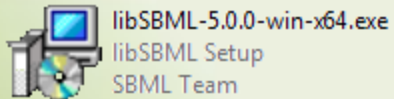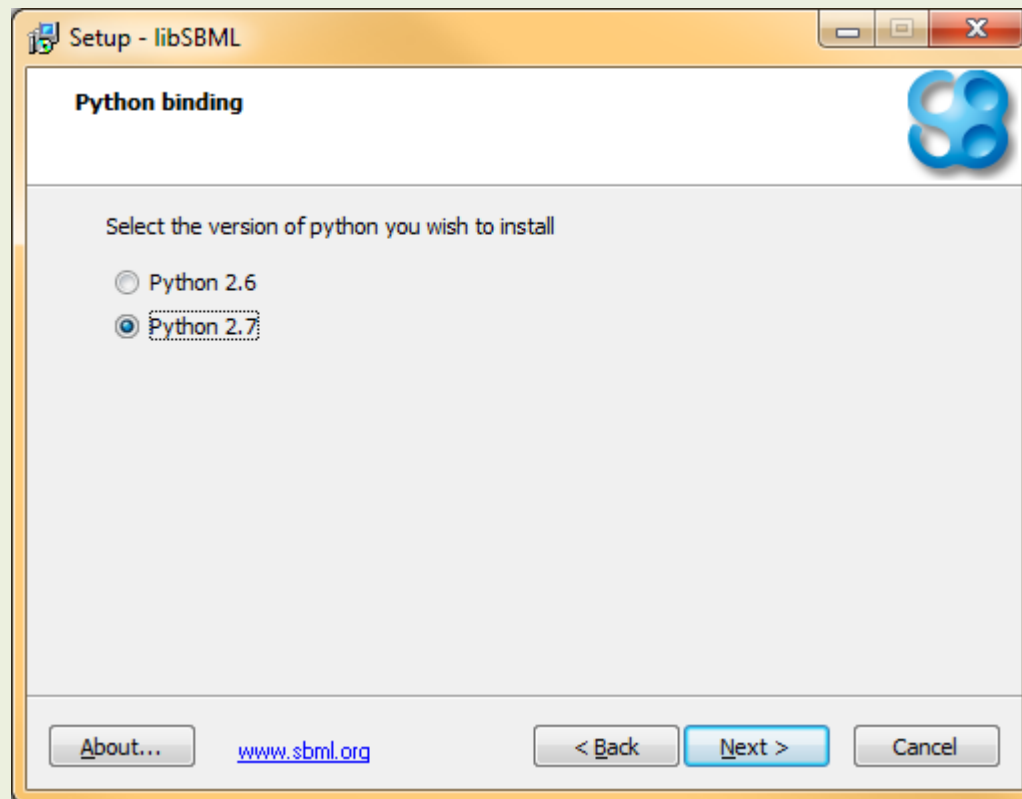
Windows Installers

Windows Installers

Windows Installers

Windows Installers
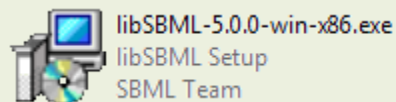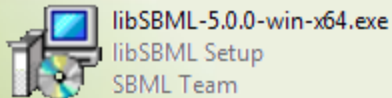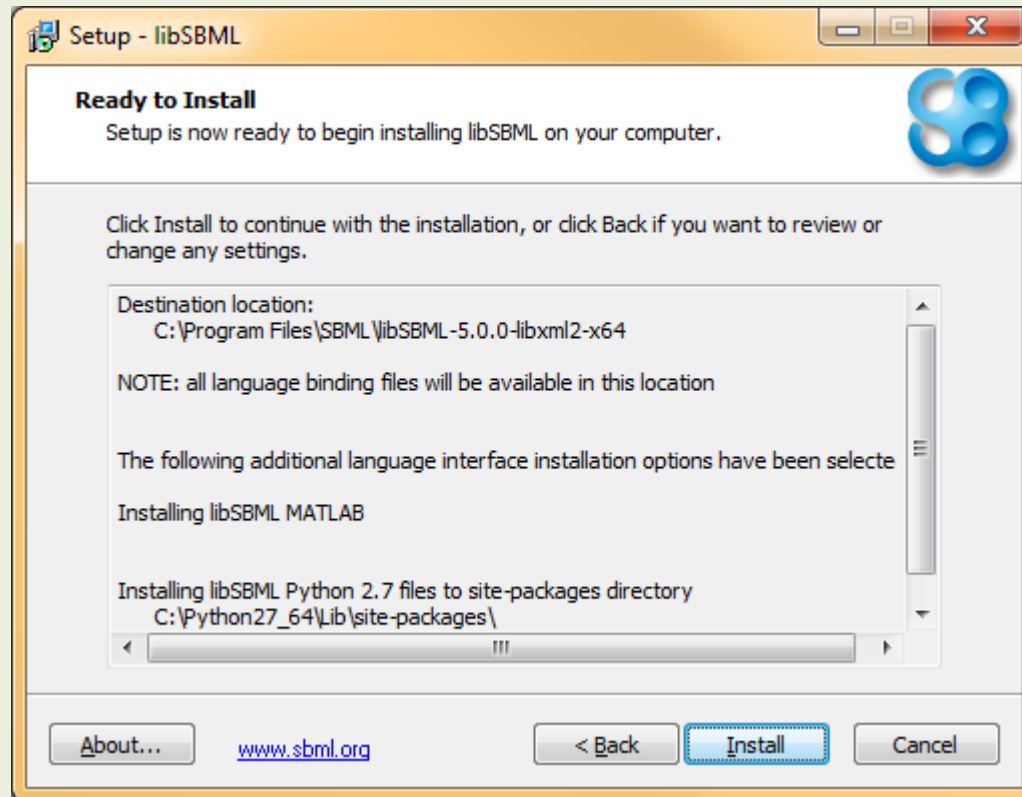
Windows Installers
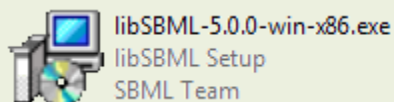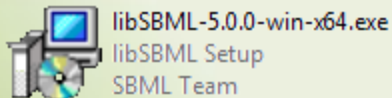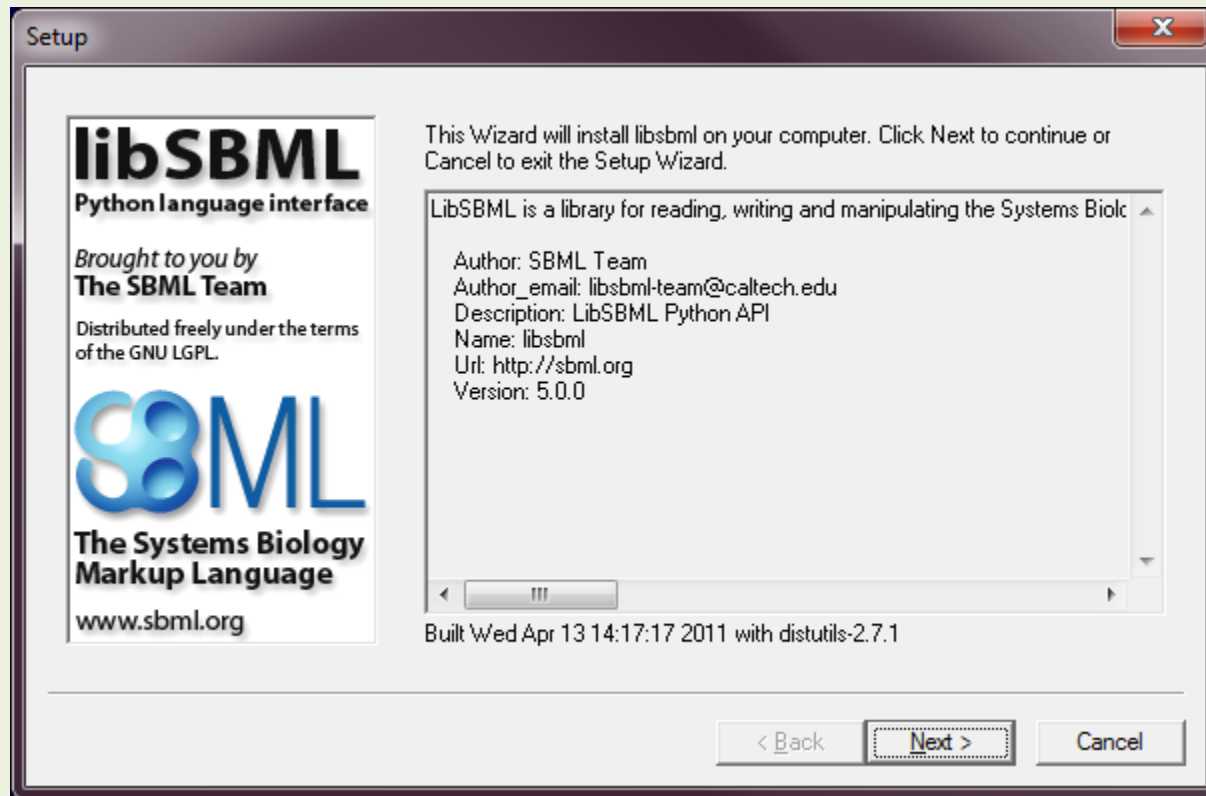
Windows Installers
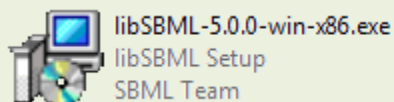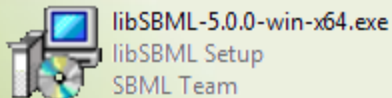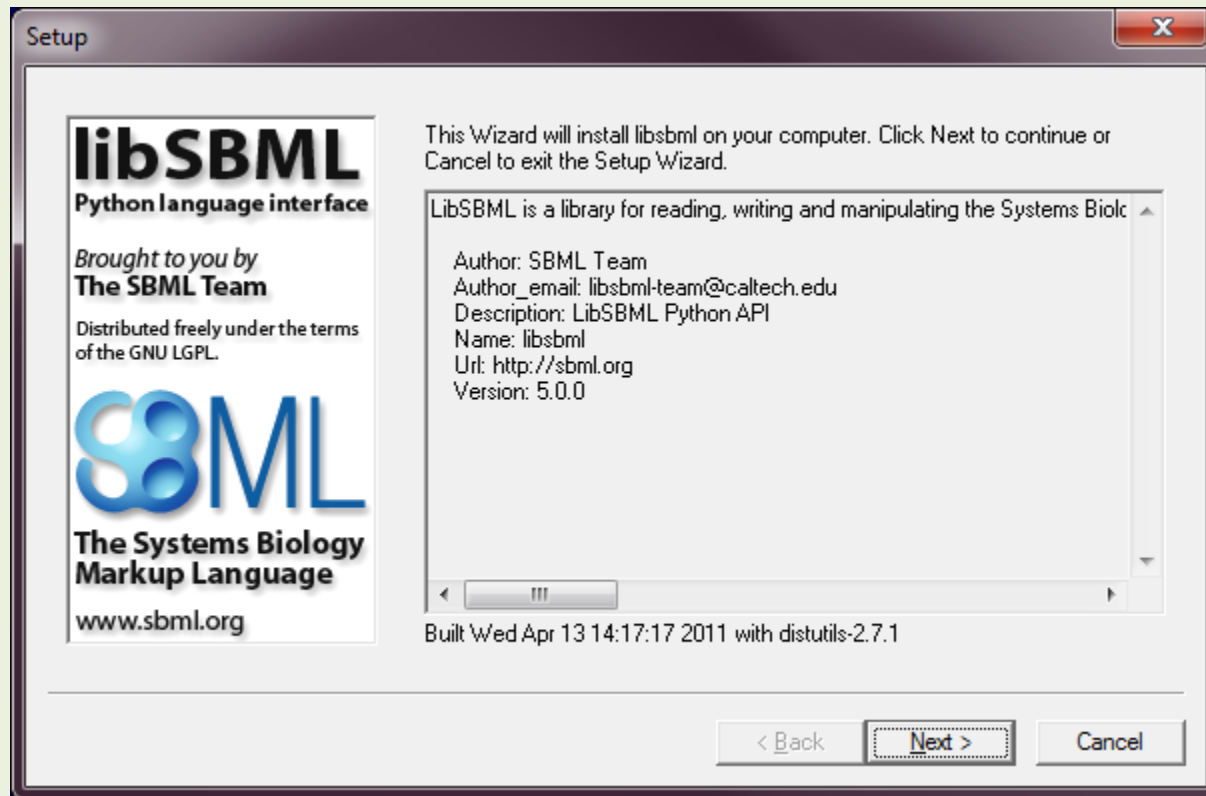
Windows Installers
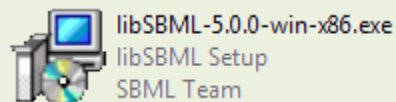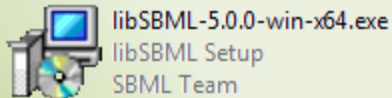
Windows Installers
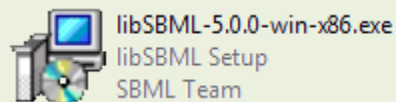
Windows Installers
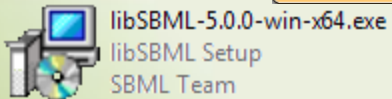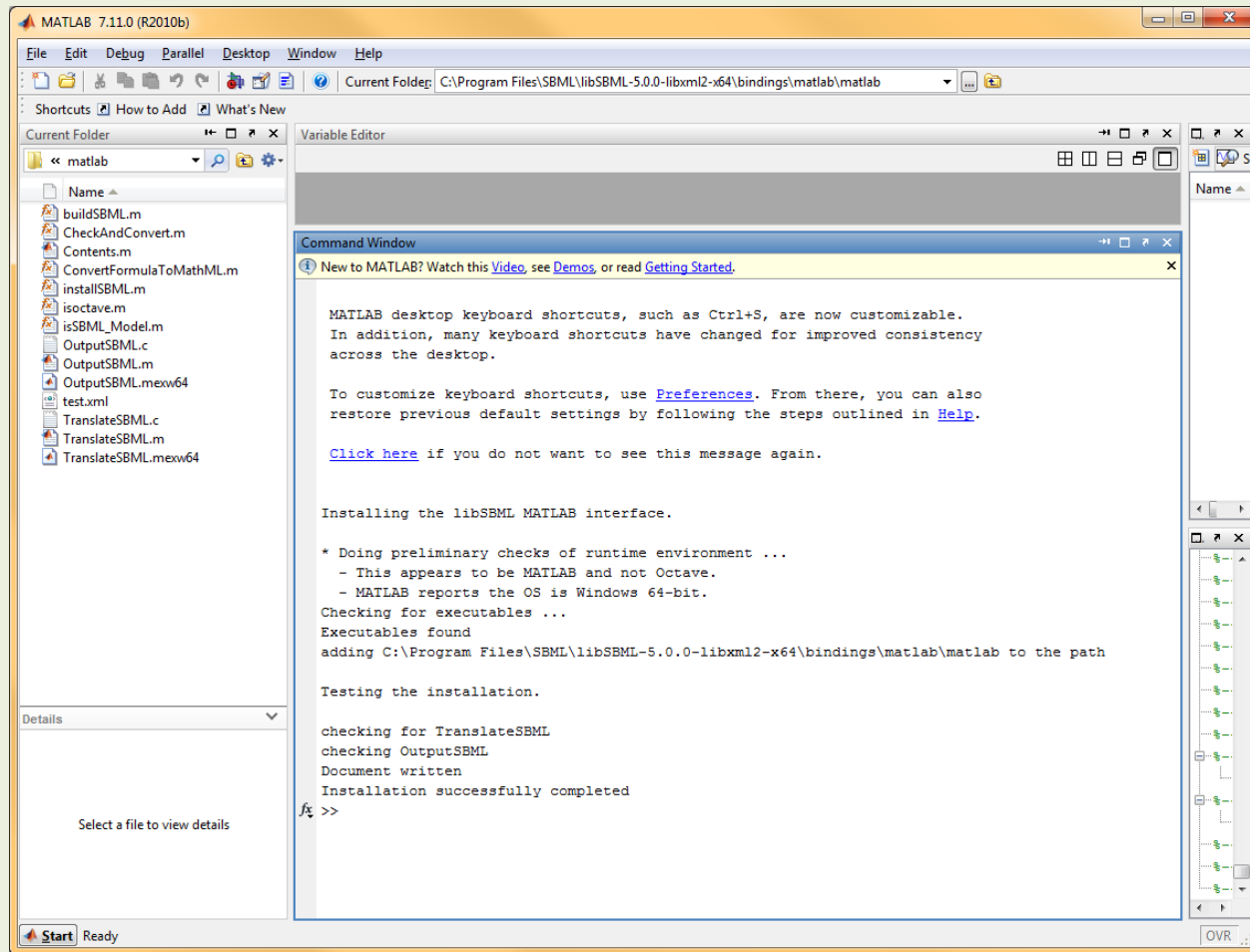
Once installed

- Language bindings for:

  - Matlab
  - Perl
  - Python

  Are available as soon as the interpreter is started. For other bindings some changes are needed ...

Once installed – C++

- INCLUDE

  ```
  C:\Program Files\SBML\libSBML-5.0.0-libxml2-x64\win64\include
  C:\Program Files\SBML\libSBML-5.0.0-libxml2-x86\win32\include
  ```

- LIB

  ```
  C:\Program Files\SBML\libSBML-5.0.0-libxml2-x64\win64\lib
  C:\Program Files\SBML\libSBML-5.0.0-libxml2-x86\win32\lib
  ```

- Remember to copy DLL files from the binary folder to your application.

Once installed – Java

- Include Jar file in your CLASSPATH
    ```
    C:\Program Files\SBML\libSBML-5.0.0-libxml2-x64\
                    bindings\java\libsbmlj.jar
    C:\Program Files\SBML\libSBML-5.0.0-libxml2-x86\
                    bindings\java\libsbmlj.jar
    ```

- Have the native library in the java.library.path
    ```
    C:\Program Files\SBML\libSBML-5.0.0-libxml2-x64\
                        bindings\java\sbmlj.dll
    C:\Program Files\SBML\libSBML-5.0.0-libxml2-x86\
                        bindings\java\sbmlj.dll
    ```

Once installed – C#

- ## Reference managed assembly:

```
C:\Program Files\SBML\libSBML-5.0.0-libxml2-x64\
                    bindings\csharp\libsbmlcsP.dll
C:\Program Files\SBML\libSBML-5.0.0-libxml2-x86\
                    bindings\csharp\libsbmlcsP.dll
```

- ## Copy the native DLL next to your executable

```
C:\Program Files\SBML\libSBML-5.0.0-libxml2-x64\
                    bindings\csharp\libsbmlcs.dll
C:\Program Files\SBML\libSBML-5.0.0-libxml2-x86\
                    bindings\csharp\libsbmlcs.dll
```

OS X Installers

OS X Installers

libSBML-5.0.0-libxml2-
snowleopard.pkg

OS X Installers

libSBML-5.0.0-libxml2-
snowleopard.pkg

OS X Installers

OS X Installers

OS X Installers – install location

libSBML–5.0.0–libxml2–
snowleopard.pkg

OS X Installers – install location

libSBML-5.0.0-libxml2-
snowleopard.pkg

OS X Installers

libSBML-5.0.0-libxml2-
snowleopard.pkg

OS X Installers

libSBML-5.0.0-libxml2-
snowleopard.pkg

OS X Installers

OS X Installers

libSBML-5.0.0-libxml2-
snowleopard.pkg
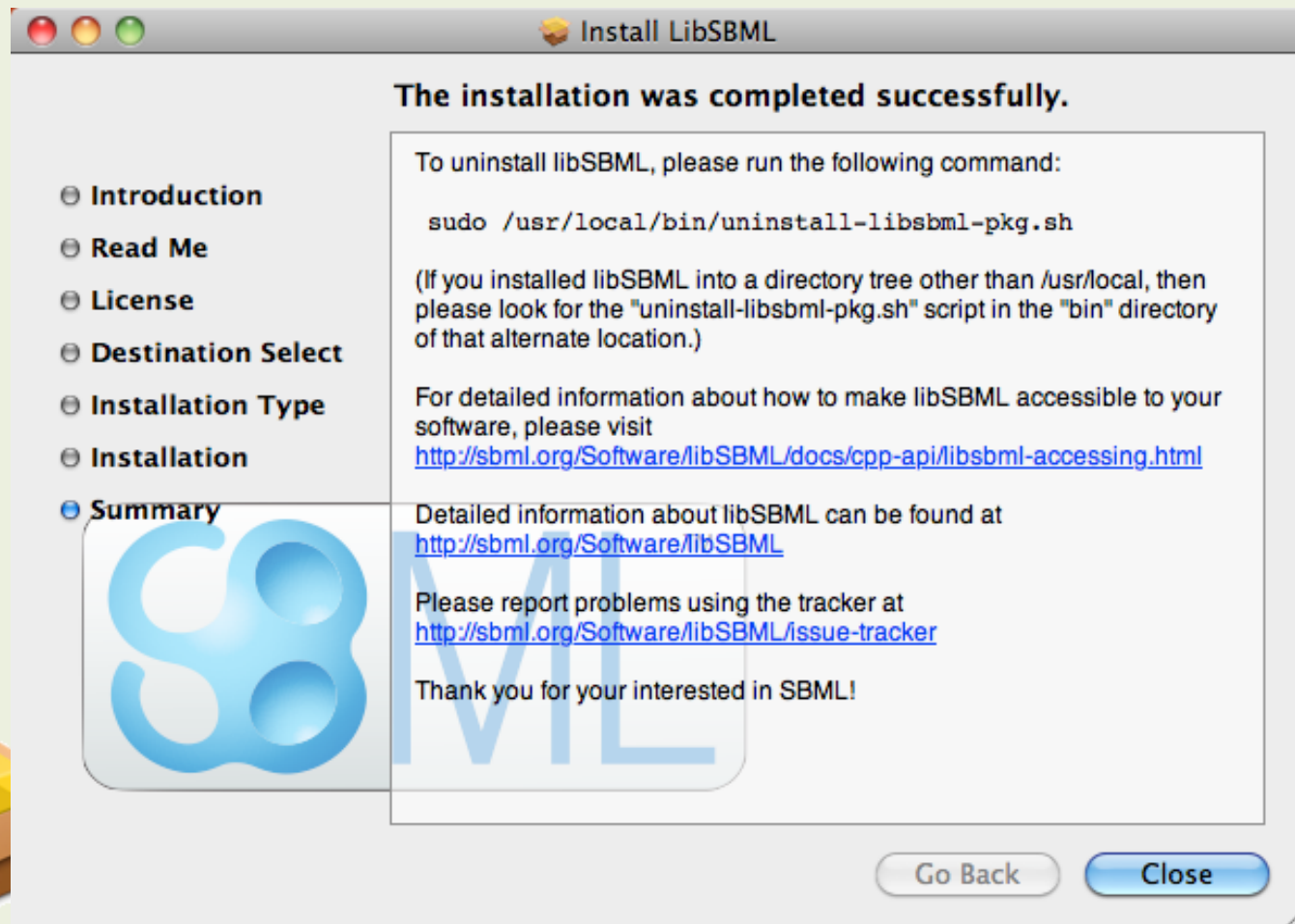
Once installed

- Language bindings for:

  - Perl
  - Python
  - Ruby

  Are available as soon as the interpreter is started.
  For other bindings some changes are needed ...

Once Installed – C++

- By default the installation directory is

  `/usr/local/`

- So one would compile with the options

`-I/usr/local/include –L/usr/local/lib -lsbml`

libSBML-5.0.0-libxml2-
snowleopard.pkg

Once installed – Java

- Include Jar file in your CLASSPATH

  `/usr/local/share/java/libsbml.jar`

- Have the native library in the java.library.path

  `/usr/local/lib/libsbmlj.jnilib`

Once installed – C#

- Reference managed assembly:
  ```
  /usr/local/lib/mono/libsbmlcsP/libsbmlcsP.dll
  ```

- Ensure that the native library is in your applications DYLD_LIBRARY_PATH
  ```
  /usr/local/lib/mono/libsbmlcsP/libsbmlcs.dylib
  /usr/local/lib/mono/libsbmlcsP/libsbml.dylib
  ```

Linux packages

- Currently only the C/C++ library is available in the DEB / RPM file.

- Have been created with:
  - Ubuntu 8.04 (DEB)
  - CentOS 4.8   (RPM)

Linux packages
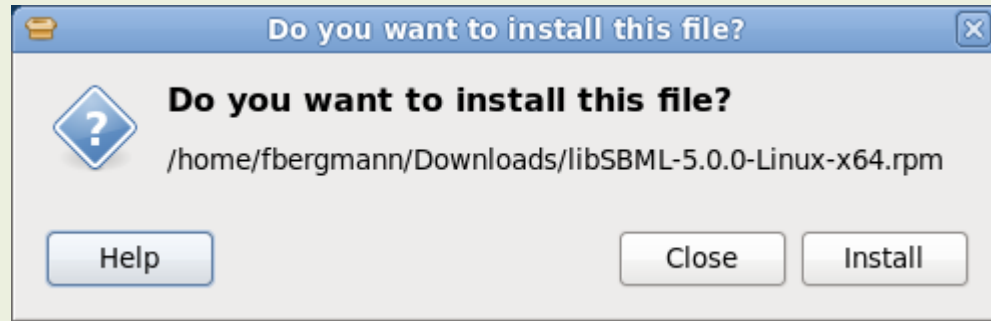
Linux packages

Once Installed – C++

- By default the installation directory is
  `/usr/local/`

- So one would compile with the options

`-I/usr/local/include –L/usr/local/lib -lsbml`

Using GNUmake / CMake

BUILDING LIBSBML

Building LibSBML

- The installers are created with compression support and using the libXML parser library.

- If the installer does not include your language bindings, or you would like to choose a different parser library you can create your own version.

Prerequisites

- XML Parser library:
  - libXML   >= 2.7.3 ⎤
  - Expat   >= 1.95.8 ⎬ choose one,
  - Xerces-c   >= 2.7.0 ⎦ (default: libXML2)
- Compression library:
  - Zlib   >= 1.2.3   (optional)
  - Bzip2   >= 1.0.5   (optional)
- Language bindings:
  - Swig / Swigwin   >= 2.0.0   (optional)
- Documentation
  - Doxygen   >= 1.6.3   (optional)
- CMake   >= 2.8.4   (optional)

GNUmake

- On Cygwin / OS X / Linux we still supply GNU make files which can be used directly by running:

```
./configure <options>
make
sudo make install
```

GNUmake options

- Installation Prefix
  `--prefix=<directory>`
- Layout Package
  `--enable-layout`
- XML Parsers
  `--with-expat[=prefix]`
  `--with-libxml[=prefix]`
  `--with-xerces[=prefix]`
- Compression
  `--with-zlib[=prefix]`
  `--with-bzip2[=prefix]`

- Language Bindings
  `--with-swig[=prefix]`
  `--with-csharp[=prefix]`
  `--with-java[=prefix]`
  `--with-octave[=prefix]`
  `--with-matlab[=prefix]`
  `--with-perl[=prefix]`
  `--with-python[=prefix]`
  `--with-ruby[=prefix]`
- Documentation
  `--with-doxygen[=prefix]`

CMake

- LibSBML is beginning to use CMake to make it easy to integrate new packages.
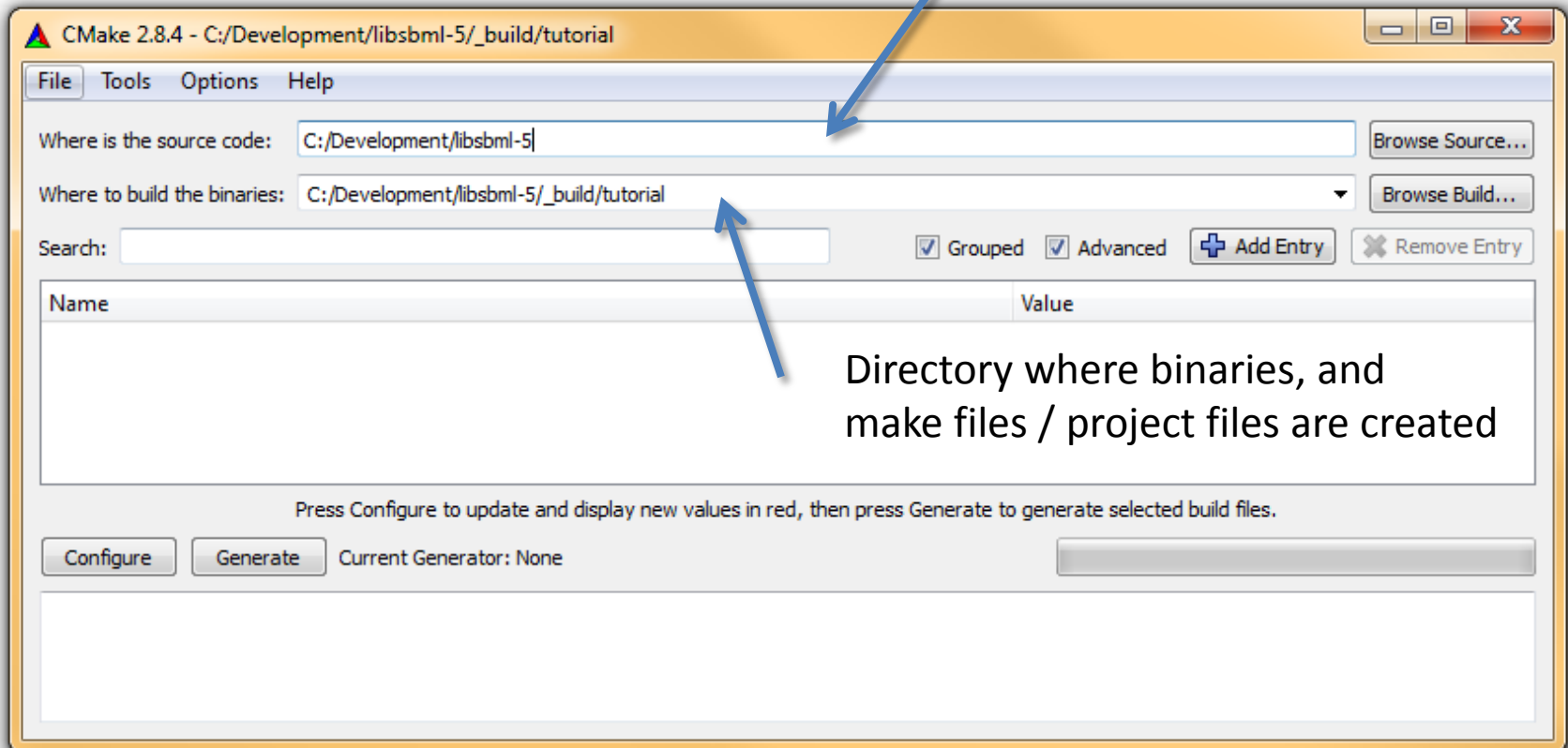
- CMake allows to generate not only make files, but also project files for commonly used IDEs:
  – Code::Blocks, eclipse (CDT), Visual Studio, Xcode
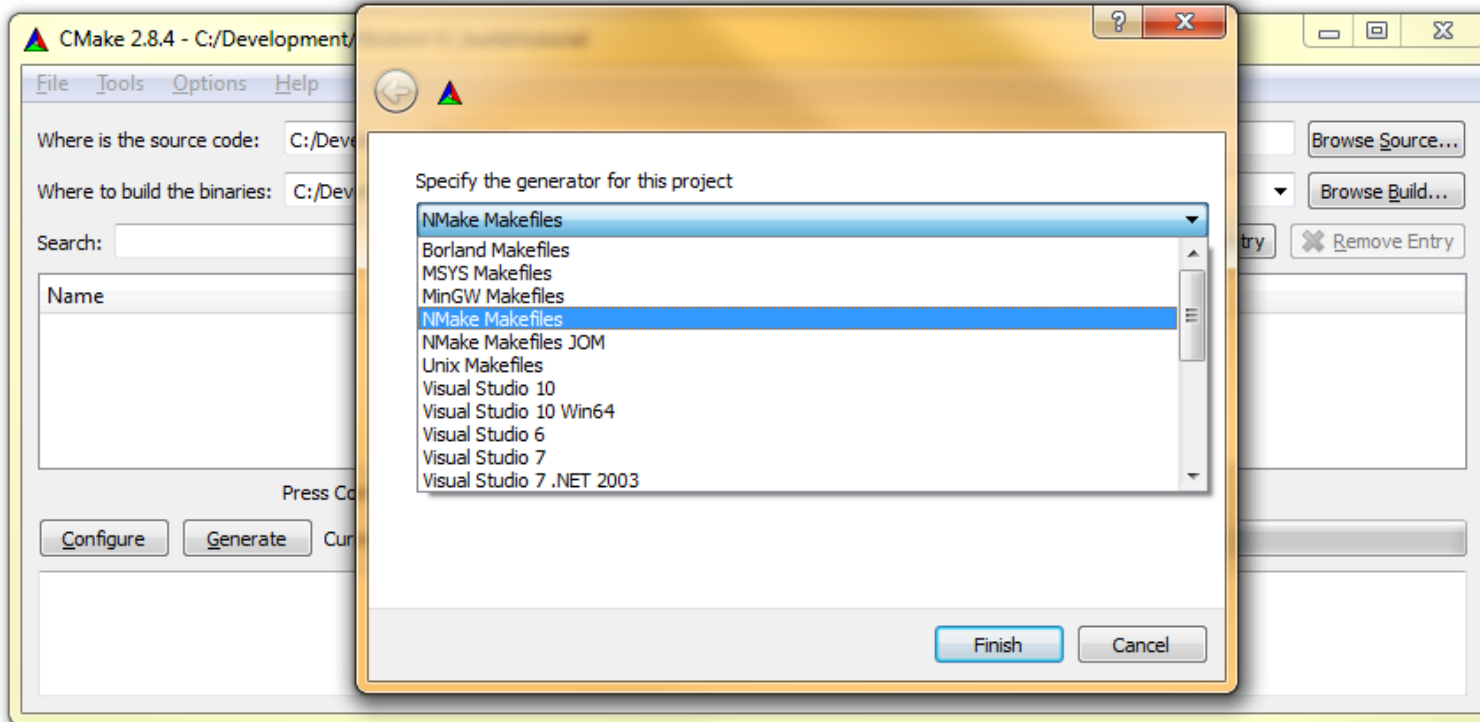
CMake configuration
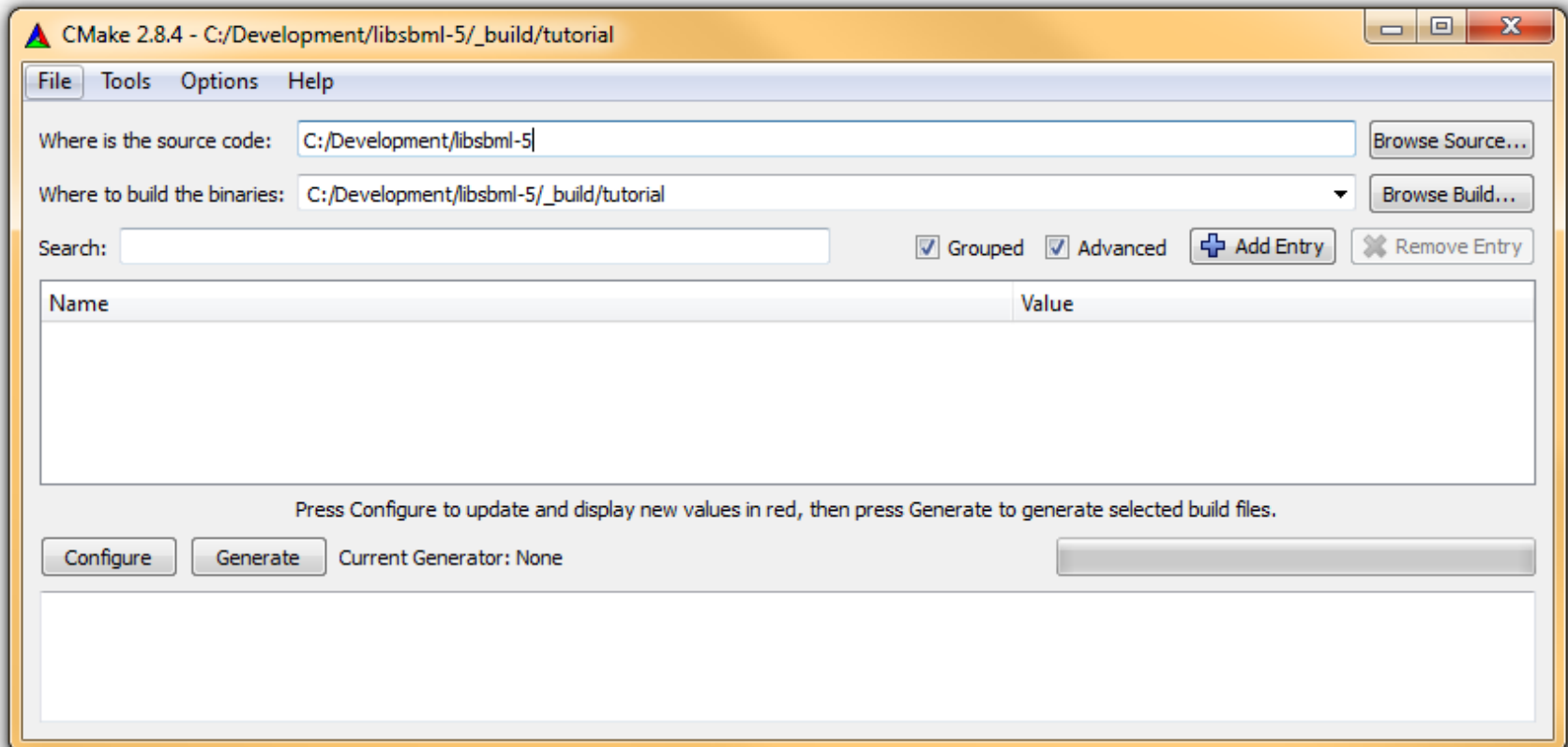
CMake configuration

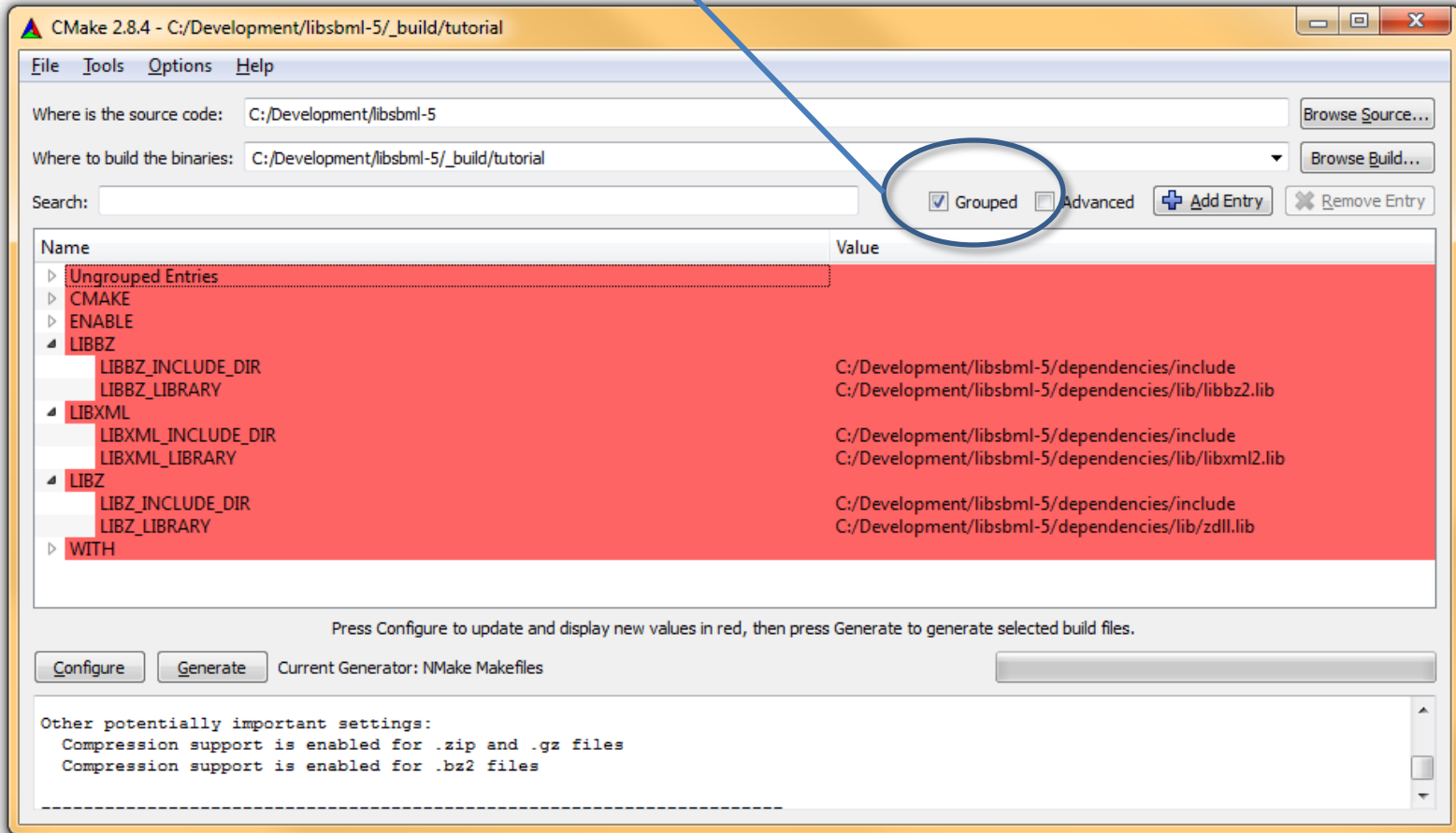

Directory with libSBML 5 sources

Directory where binaries, and make files / project files are created
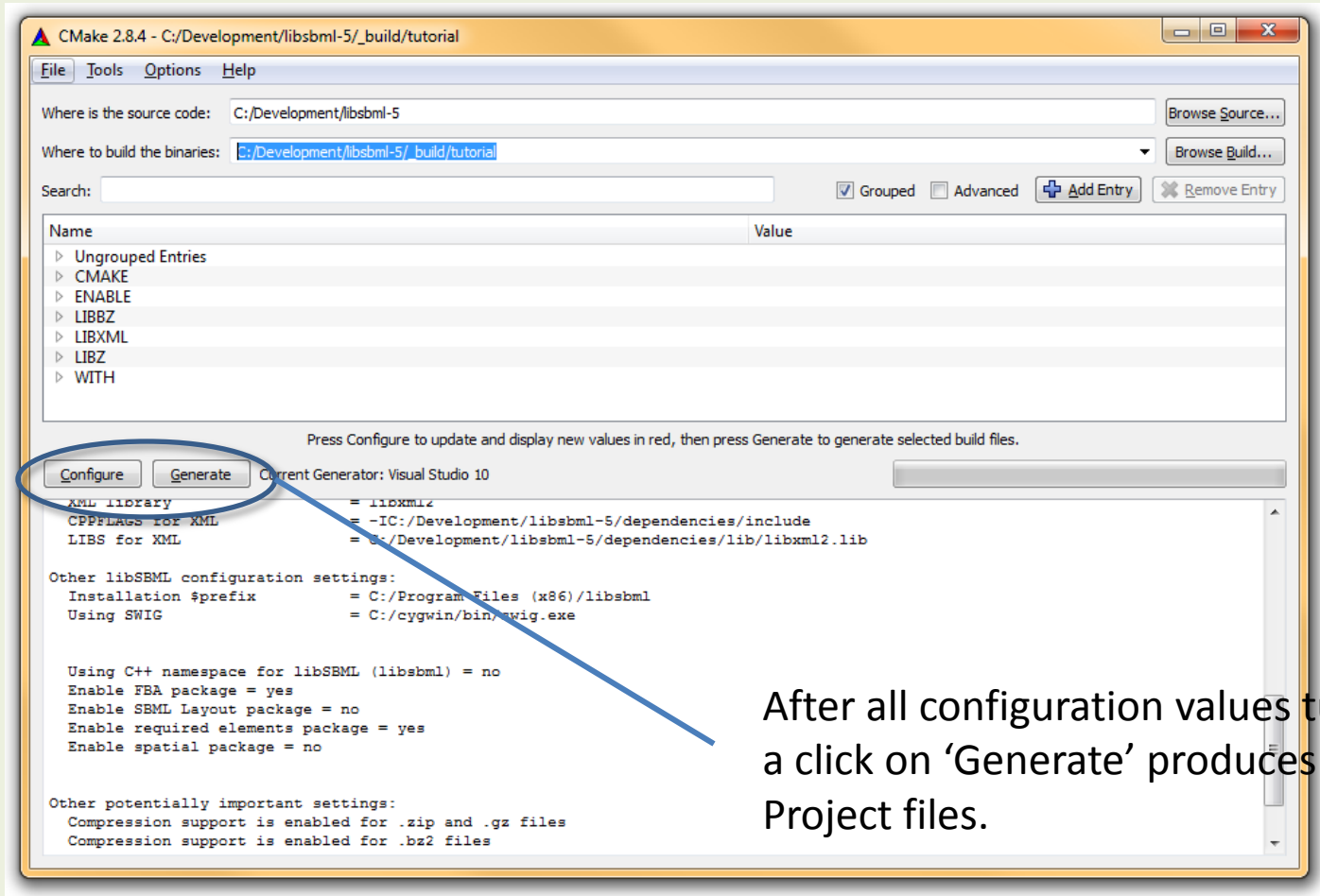
CMake configuration

CMake configuration

Grouping makes it easy to navigate the options

CMake configuration

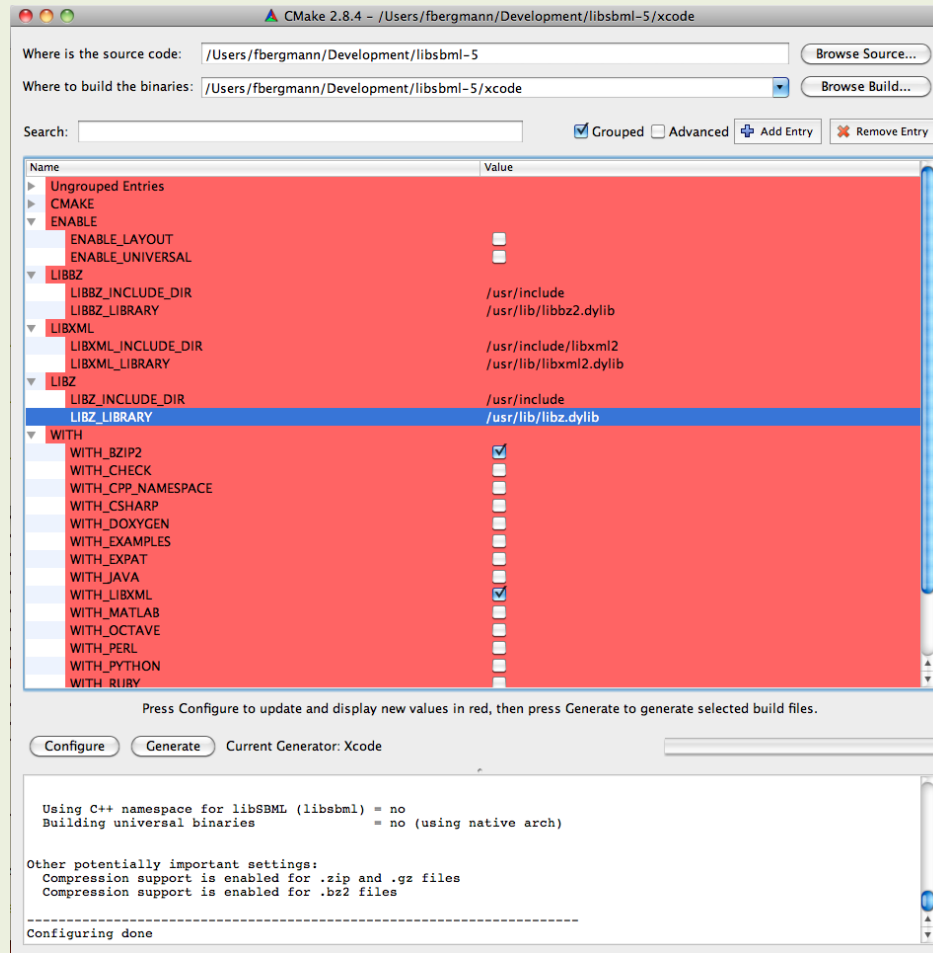After all configuration values turned white a click on 'Generate' produces the Project files.
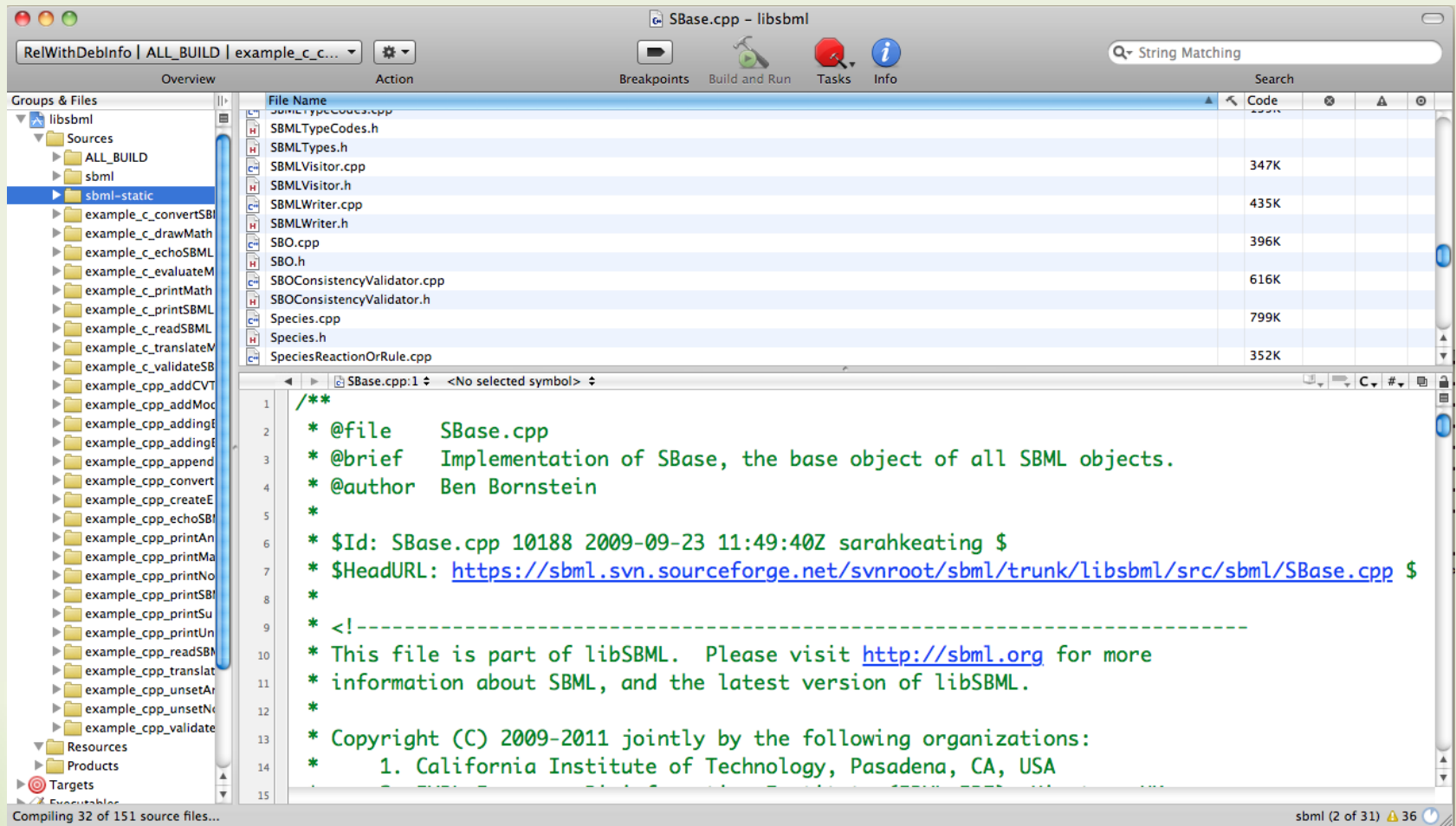
Visual Studio
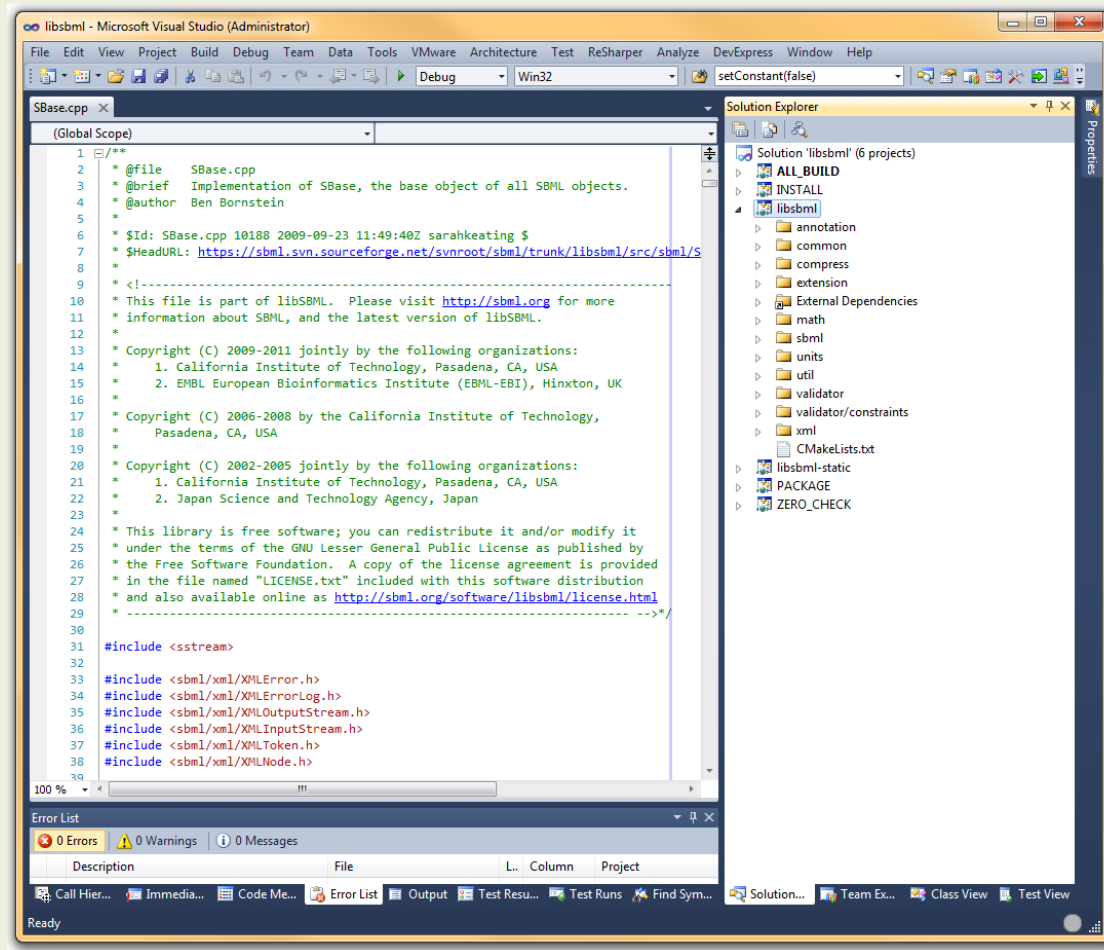
CMake configuration on OS X

CMake configuration on OS X

CMake configuration

- CMake can generate the make files / project files also from the command line:

```
cmake –DENABLE_LAYOUT=ON –DWITH_EXAMPLES=ON
          <libsbml source directory>
```

  Would configure the build with layout extension and examples.

CMake options

- Installation Prefix

  `CMAKE_INSTALL_PREFIX=<directory>`

- Layout Package

  `WITH_LAYOUT=ON`

- XML Parsers

  `WITH_EXPAT=ON`

  `WITH_LIBXML=ON`

  `WITH_XERCES=ON`

- Compression

  `WITH_ZLIB=ON`

  `WITH_BZIP2=ON`

- Language Bindings

  `WITH_SWIG=ON`

  `WITH_CSHARP=ON`

  `WITH_JAVA=ON`

  `WITH_OCTAVE=ON`

  `WITH_MATLAB=ON`

  `WITH_PERL=ON`

  `WITH_PYTHON=ON`

  `WITH_RUBY=ON`

- Documentation

  `WITH_DOXYGEN=ON`

Installing

- GNU make: sudo make install

- CMake Unix Makefile project: sudo make install

- CMake NMake project: nmake install

- CMake IDE project: select the INSTALL target of your Release configuration and build it.

DOCUMENTATION

http://sbml.org/Software/libSBML/

Acknowledgements

Akiya Jouraku

Keio, Japan

Ben Bornstein

JPL, USA

Mike Hucka

Caltech, USA