

CMake Trilinos?

Roscoe A. Bartlett

<http://www.cs.sandia.gov/~rabartl/>

Department of Optimization & Uncertainty Estimation

Esteban J. Guillen

Department of Information Engineering

Sandia National Laboratories

Trilinos User Group Meeting, October 21, 2008

- What is CMake?
- User advantages in switching Trilinos to CMake
- Configuring, building, and installing Trilinos with CMake on Unix systems
- Native Microsoft Windows support
 - Self-extracting installer for Trilinos
 - Visual C++ project files and Windows CMake GUI

- CMake = “Cross-platform Make”
- CMake:
 - Build system primarily for C/C++ code
 - Front-ends to configure a software package
 - Command-line, Scripts, CURSES, GUIs
 - Back-ends that build code
 - Unix Makefiles, MS Visual C++ Projects, Eclipse Projects, ...
 - Packaging and installing
 - Tar/gzip, Windows self-extracting installers, PackageManager, RPM, ...
- Platforms and usage:
 - Platforms:
 - Unix/Linux, MAC OSX, MS Windows, AIX, IRIX, ...
 - Internal Sandia use:
 - VTK/Titan, ParaView, ThreatView, ...
 - External use:
 - KDE, MySql, MiKTeX, (and many many more) ..

CMake is a full featured mature build system!



User advantages in switching Trilinos to CMake?

- Provide native support for MS Windows
 - Visual C++ projects
 - GUI binary installers
- Better user feedback for configuration errors
- Better support for shared libraries on many platforms
- More packaging and installation options
- Easier configuration for complex package dependencies

- Our detailed evaluation of CMake for Trilinos is finished:
 - Roscoe A. Bartlett, Daniel Dunlavy, Guillen Esteban, and Tim Shead. *Trilinos CMake Evaluation*. SAND2008-xxxx, October 2008
 - <http://www.cs.sandia.gov/~rabartl/publications.html>
- We have a nearly complete CMake build system design in Trilinos Dev
- Current CMake enabled packages:
 - Teuchos, RTOp, Epetra, Triutils, EpetraExt, Thyra, RBGen
- Trilinos community close to making a decision to move to CMake?

Configuring Trilinos with CMake on Unix/Linux

- CMake interactive mode: **[Not Recommended]**

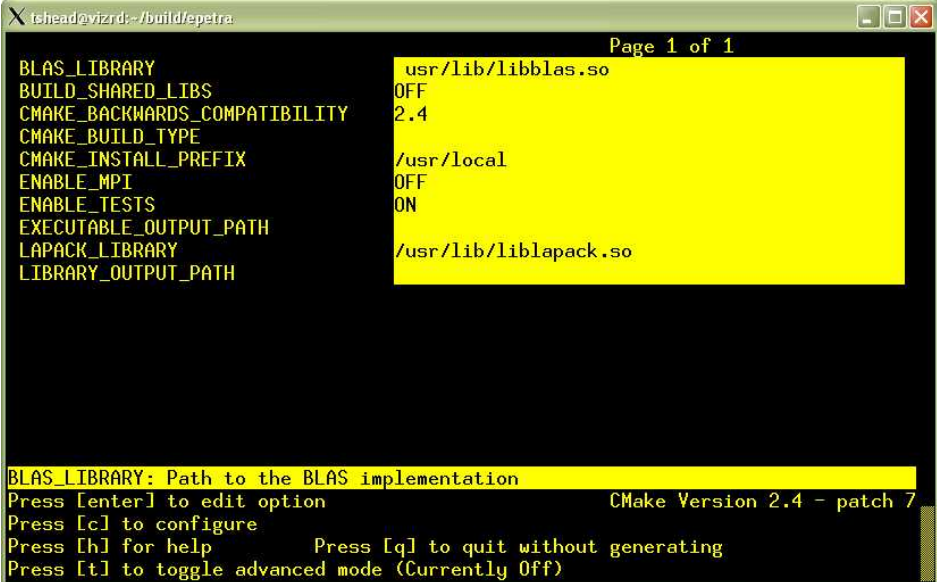
```
$ cmake -i $TRILINOS_HOME
```

- CCMake (CURSES):

```
$ ccmake $TRILINOS_HOME
```

- CMake script files:

```
$ cmake -S script_file \  
$TRILINOS_HOME
```



The screenshot shows a terminal window titled 'tehead@vizrd: ~/build/epetra'. The window displays the CCMake (CURSES) interface for configuring Trilinos. The left pane lists configuration options, and the right pane shows their current values. The options and values are:

Option	Value
BLAS_LIBRARY	/usr/lib/libblas.so
BUILD_SHARED_LIBS	OFF
CMAKE_BACKWARDS_COMPATIBILITY	2.4
CMAKE_BUILD_TYPE	
CMAKE_INSTALL_PREFIX	/usr/local
ENABLE_MPI	OFF
ENABLE_TESTS	ON
EXECUTABLE_OUTPUT_PATH	
LAPACK_LIBRARY	/usr/lib/liblapack.so
LIBRARY_OUTPUT_PATH	

At the bottom, there is a status bar with the following text:

```
BLAS_LIBRARY: Path to the BLAS implementation  
Press [enter] to edit option  
Press [c] to configure  
Press [h] for help  
Press [q] to quit without generating  
Press [t] to toggle advanced mode (Currently Off)  
CMake Version 2.4 - patch 7
```

- CMake command-line options: **[Recommended]**

```
$ cmake -D Trilinos_ENABLE_ALL_PACKAGES:BOOL=ON \  
-D Trilinos_ENABLE_TESTS:BOOL=ON ... $TRILINOS_HOME
```

Creating a Configuration Script for CMake

```
#!/bin/sh
EXTRA_ARGS=$@
cmake \
  -D CMAKE_CXX_FLAGS:String="-g -O0 -ansi -pedantic -Wall" \
  -D DART_TESTING_TIMEOUT:String=600 \
  -D Trilinos_ENABLE_NOX:BOOL=ON \
  -D Trilinos_ENABLE_ALL_OPTIONAL_PACKAGES:BOOL=ON \
  -D Trilinos_ENABLE_EXAMPLES:BOOL=ON \
  -D Trilinos_ENABLE_TESTS:BOOL=ON \
  ... \
  $EXTRA_ARGS \
  ../../../../Trilinos
```

```
$ ./do-configure -D VEROBSE_CONFIGURE:BOOL=ON
$ make -j4
$ ctest
$ make install
```

See example scripts:

[Trilinos/sampleScripts/*cmake](#)

- Configuring Trilinos to build all packages with all tests and examples:

- D Trilinos_ENABLE_ALL_PACKAGES:BOOL=ON
 - D Trilinos_ENABLE_TESTS:BOOL=ON
 - D Trilinos_ENABLE_EXAMPLES:BOOL=ON

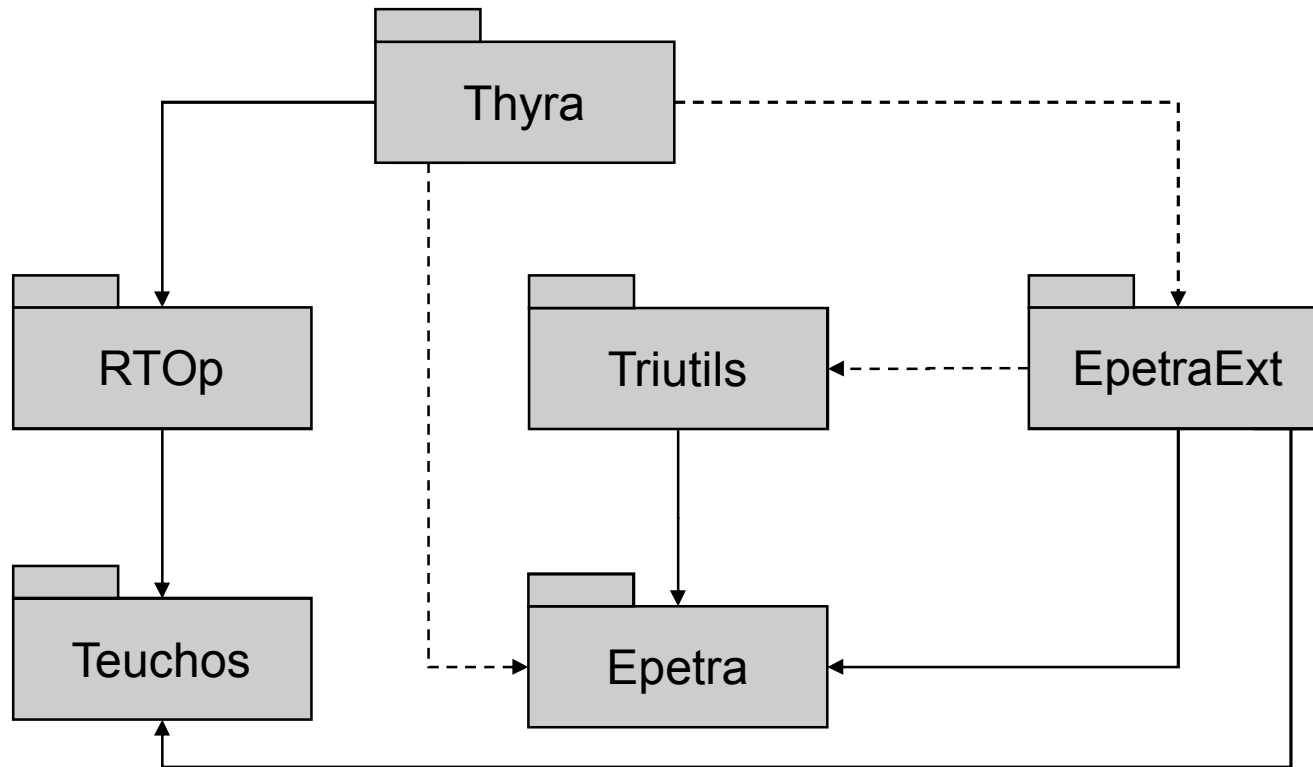
- Configuring a package(s) along with all of the packages it can use

- D Trilinos_ENABLE_Stratimikos:BOOL=ON
 - D Trilinos_ENABLE_ALL_OPTIONAL_PACKAGES:BOOL=ON
 - D Trilinos_ENABLE_TESTS:BOOL=ON
 - D Trilinos_ENABLE_EXAMPLES:BOOL=ON

- Configuring Trilinos to disable a package(s) and all packages it depends on:

- D Trilinos_ENABLE_Stratimikos:BOOL=ON
 - D Trilinos_ENABLE_Amesos:BOOL=OFF

Example: Enabling a Package and All Optional Packages



Required Dependence →

Optional Dependence - - - - ->

Example: Enabling a Package and All Optional Packages

```
$ ./do-configure -DTrilinos_ENABLE_ALL_PACKAGES:BOOL=OFF \  
-DTrilinos_ENABLE_Thyra:BOOL=ON \  
-DTrilinos_ENABLE_ALL_OPTIONAL_PACKAGES:BOOL=ON
```

Configuring Trilinos build directory

...

Enabling all optional packages for current set of enabled packages ...

```
-- Setting Trilinos_ENABLE_EpetraExt=ON because Trilinos_ENABLE_Thyra=ON  
-- Setting Trilinos_ENABLE_Epetra=ON because Trilinos_ENABLE_Thyra=ON  
-- Setting Trilinos_ENABLE_Triutils=ON because Trilinos_ENABLE_EpetraExt=ON
```

Enabling all remaining required packages for the current set of enabled packages ...

```
-- Setting Trilinos_ENABLE_RTop=ON because Trilinos_ENABLE_Thyra=ON  
-- Setting Trilinos_ENABLE_Teuchos=ON because Trilinos_ENABLE_Thyra=ON
```

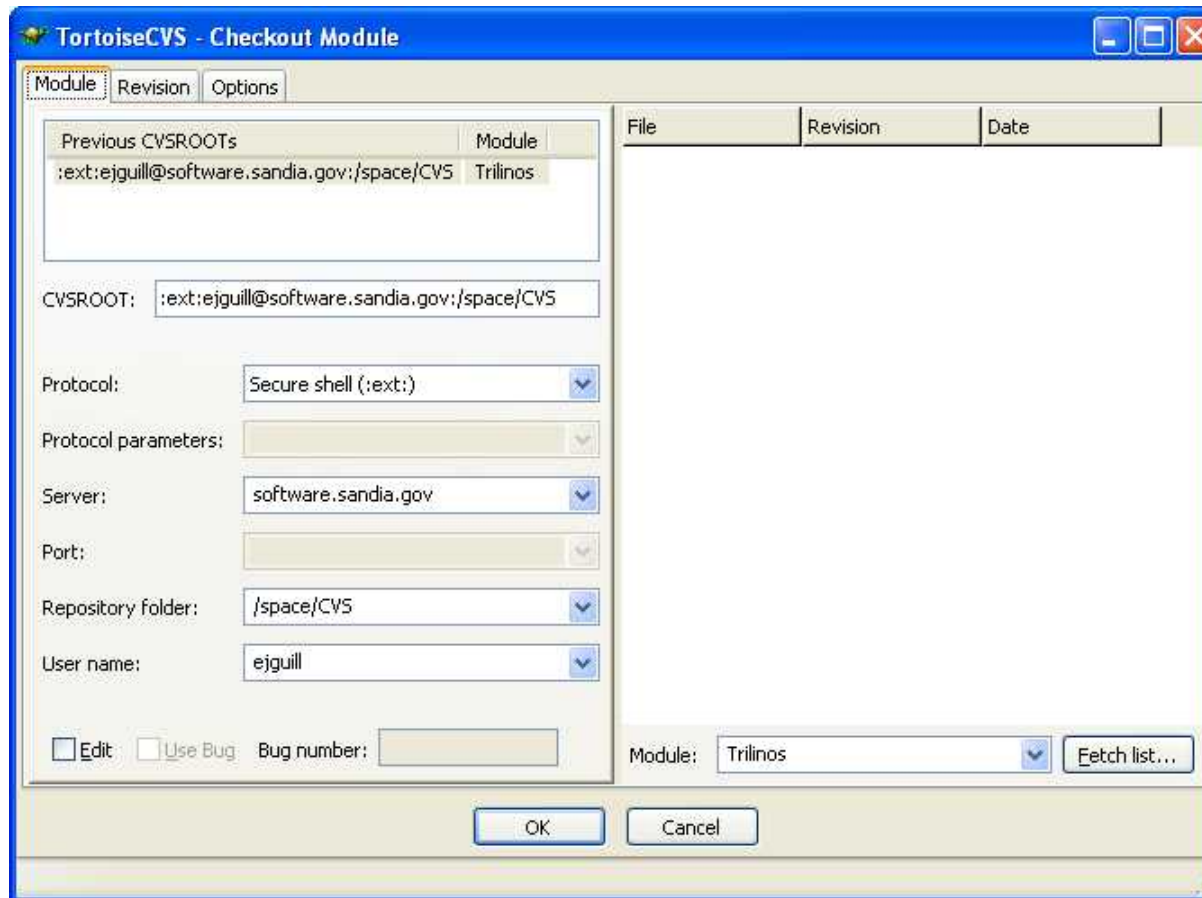
Enabling all optional intra-package enables that can be if both sets of packages are enabled ...

```
-- Setting EpetraExt_ENABLE_Triutils=ON since Trilinos_ENABLE_EpetraExt=ON AND Trilinos_ENABLE_Triutils=ON  
-- Setting Thyra_ENABLE_EpetraExt=ON since Trilinos_ENABLE_Thyra=ON AND Trilinos_ENABLE_EpetraExt=ON  
-- Setting Thyra_ENABLE_Epetra=ON since Trilinos_ENABLE_Thyra=ON AND Trilinos_ENABLE_Epetra=ON
```

Final set of enabled packages: Teuchos RTop Epetra Triutils EpetraExt Thyra 6

DEMO

Checkout Trilinos From CVS



The image shows the 'TortoiseCVS - Checkout Module' dialog box. It has three tabs: 'Module', 'Revision', and 'Options'. The 'Module' tab is active. On the left, there is a table of 'Previous CVSROOTs' with columns 'Previous CVSROOTs' and 'Module'. The first entry is ':ext:ejguill@software.sandia.gov:/space/ CVS' with 'Trilinos' in the 'Module' column. Below this table is a text field for 'CVSROOT:' containing ':ext:ejguill@software.sandia.gov:/space/ CVS'. Further down are several fields with dropdown menus: 'Protocol:' (set to 'Secure shell (:ext:)'), 'Protocol parameters:', 'Server:' (set to 'software.sandia.gov'), 'Port:', 'Repository folder:' (set to '/space/ CVS'), and 'User name:' (set to 'ejguill'). At the bottom left are checkboxes for 'Edit' and 'Use Bug', followed by a 'Bug number:' field. On the right side of the dialog is a large table with columns 'File', 'Revision', and 'Date'. At the bottom right, there is a 'Module:' dropdown set to 'Trilinos' and a 'Fetch list...' button. At the very bottom are 'OK' and 'Cancel' buttons.

Previous CVSROOTs	Module
:ext:ejguill@software.sandia.gov:/space/ CVS	Trilinos

CVSROOT: :ext:ejguill@software.sandia.gov:/space/ CVS

Protocol: Secure shell (:ext:)

Protocol parameters:

Server: software.sandia.gov

Port:

Repository folder: /space/ CVS

User name: ejguill

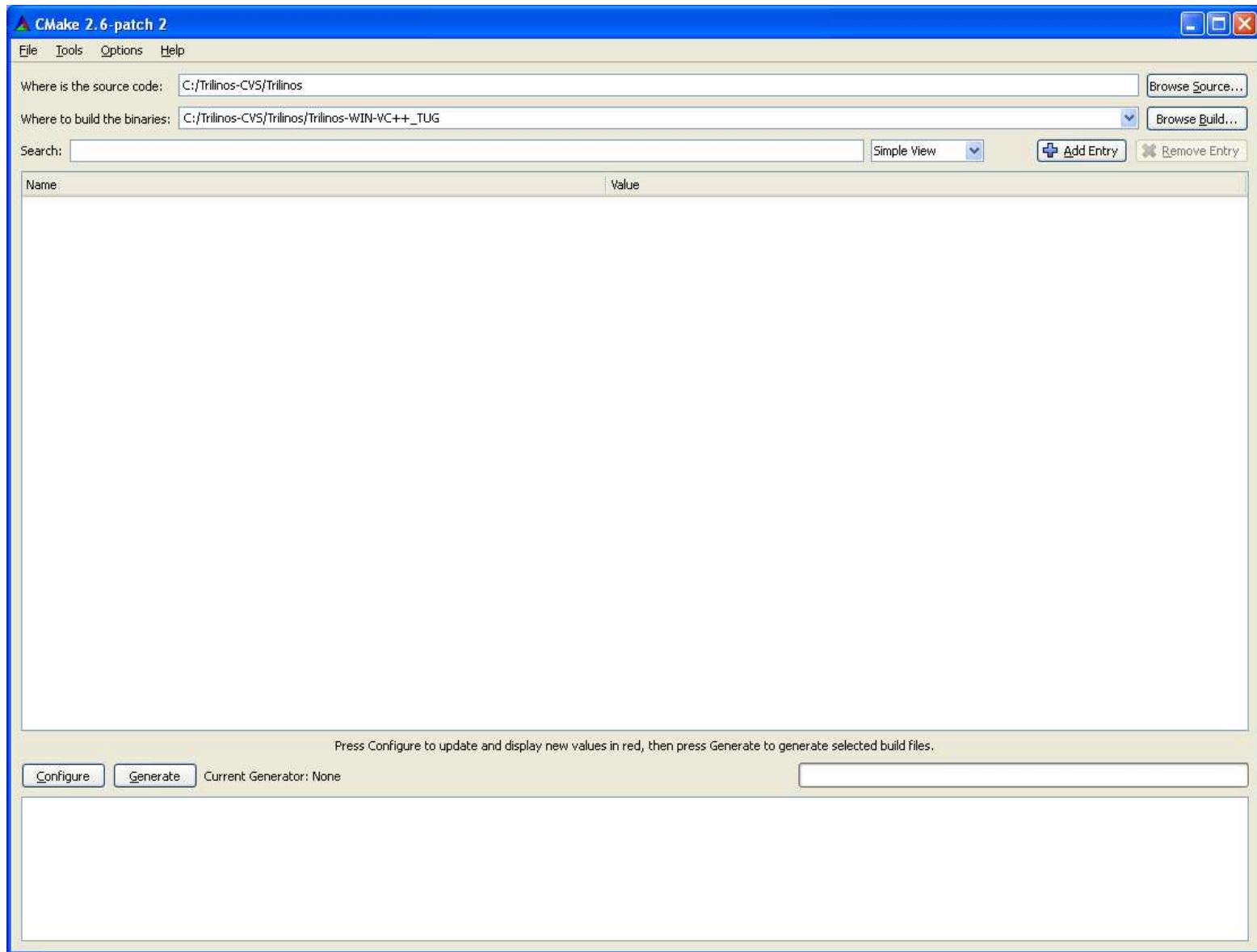
☐ Edit ☐ Use Bug Bug number:

File	Revision	Date
------	----------	------

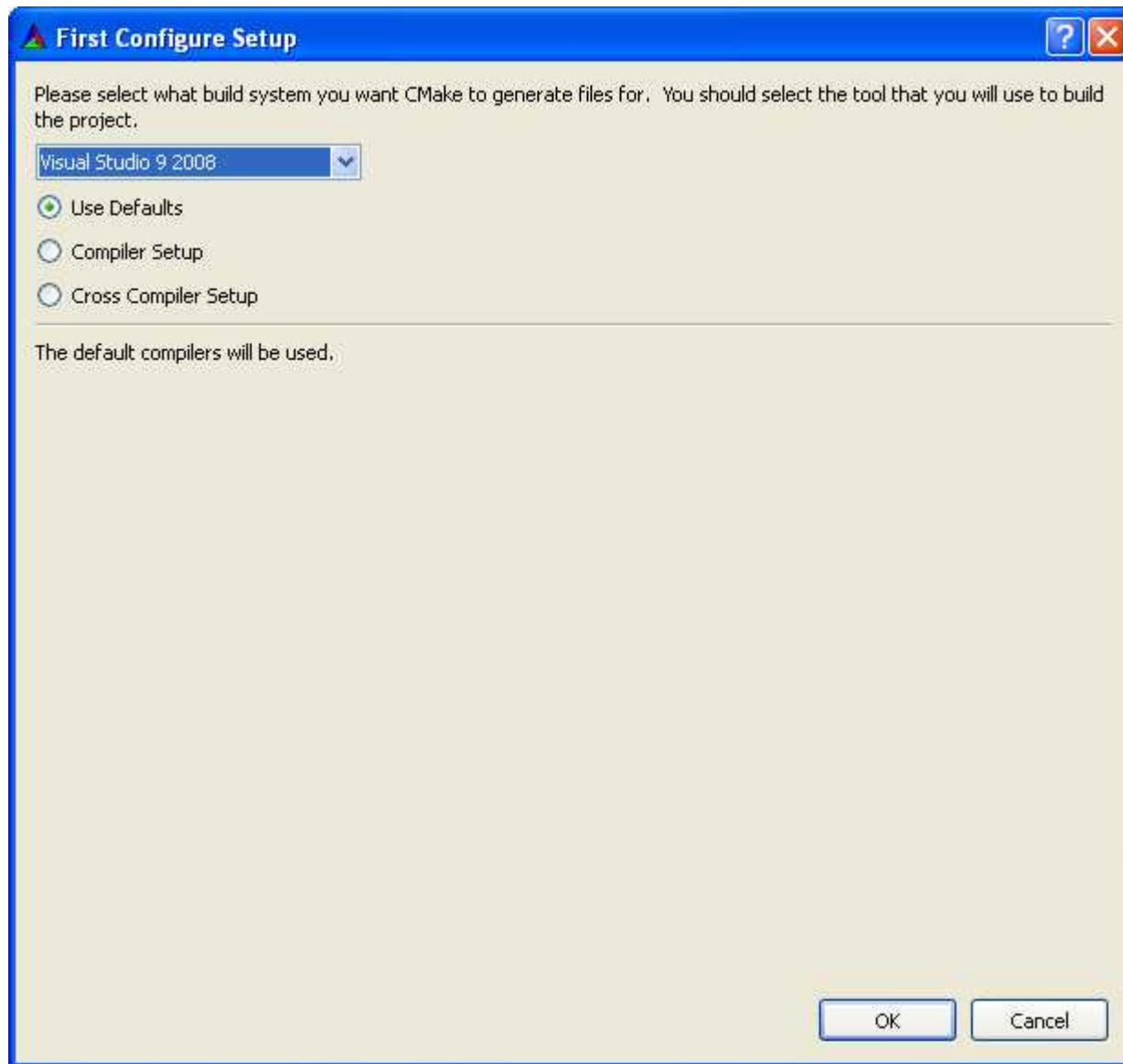
Module: Trilinos Fetch list...

OK Cancel

Open The CMake GUI



Define The Generator



Configure

CMake 2.6-patch 2

File Tools Options Help

Where is the source code:

Where to build the binaries:

Search:

Name	Value
PERL_EXECUTABLE	C:/Cygwin/bin/perl.exe
RBGen_ENABLE_EXAMPLES	
RBGen_ENABLE_Epetra	
RBGen_ENABLE_TESTS	
RTop_ENABLE_EXAMPLES	
RTop_ENABLE_TESTS	
SCPCOMMAND	C:/Cygwin/bin/scp.exe
SITE	S893218
SVNCOMMAND	C:/Cygwin/bin/svn.exe
TRILINOS_BLAS_LIBRARY	C:/WINDOWS/system32/blas_win32.lib
TRILINOS_ENABLE_MPI	☐
TRILINOS_HOSTNAME	S893218
TRILINOS_LAPACK_LIBRARY	C:/WINDOWS/system32/lapack_win32.lib
Teuchos_ENABLE_EXAMPLES	
Teuchos_ENABLE_TESTS	
Thyra_ENABLE_EXAMPLES	
Thyra_ENABLE_Epetra	
Thyra_ENABLE_EpetraExt	
Thyra_ENABLE_TESTS	
Trilinos_ENABLE_ALL_FORWARD_DEP_PACKAGES	☐
Trilinos_ENABLE_ALL_OPTIONAL_PACKAGES	☐
Trilinos_ENABLE_ALL_PACKAGES	☒
Trilinos_ENABLE_DEPENDENCY_UNIT_TESTS	☐
Trilinos_ENABLE_EXAMPLES	
Trilinos_ENABLE_Epetra	
Trilinos_ENABLE_EpetraExt	
Trilinos_ENABLE_FORTRAN	☐
Trilinos_ENABLE_NATIVE_TEST_HARNESS	☐
Trilinos_ENABLE_RBGen	
Trilinos_ENABLE_RTop	☐
Trilinos_ENABLE_TESTS	
Trilinos_ENABLE_Teuchos	
Trilinos_ENABLE_Thyra	

Enable the native Trilinos perl-based test harness.

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 9 2008

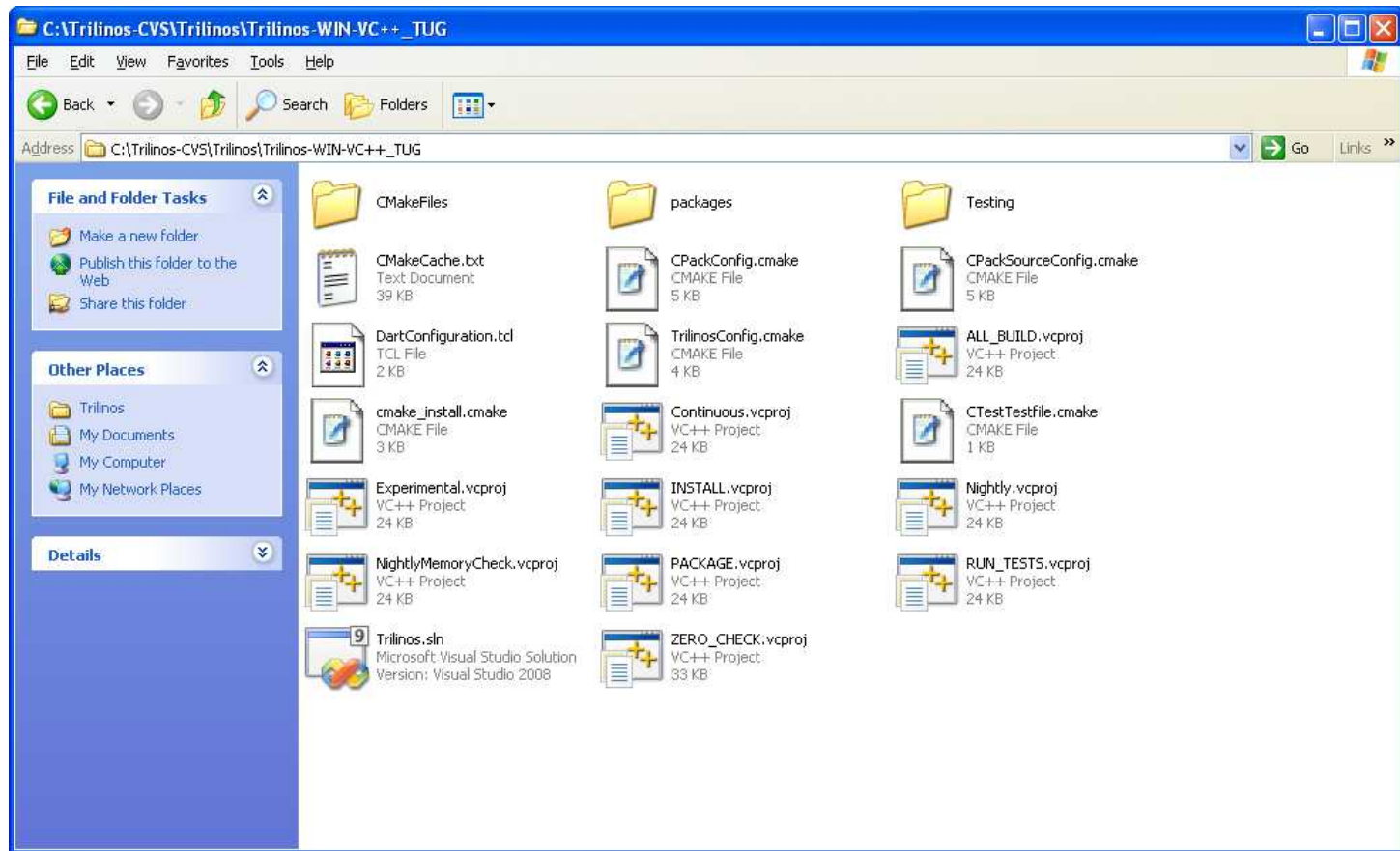
Final set of enabled packages: 0

Probing the environment ...

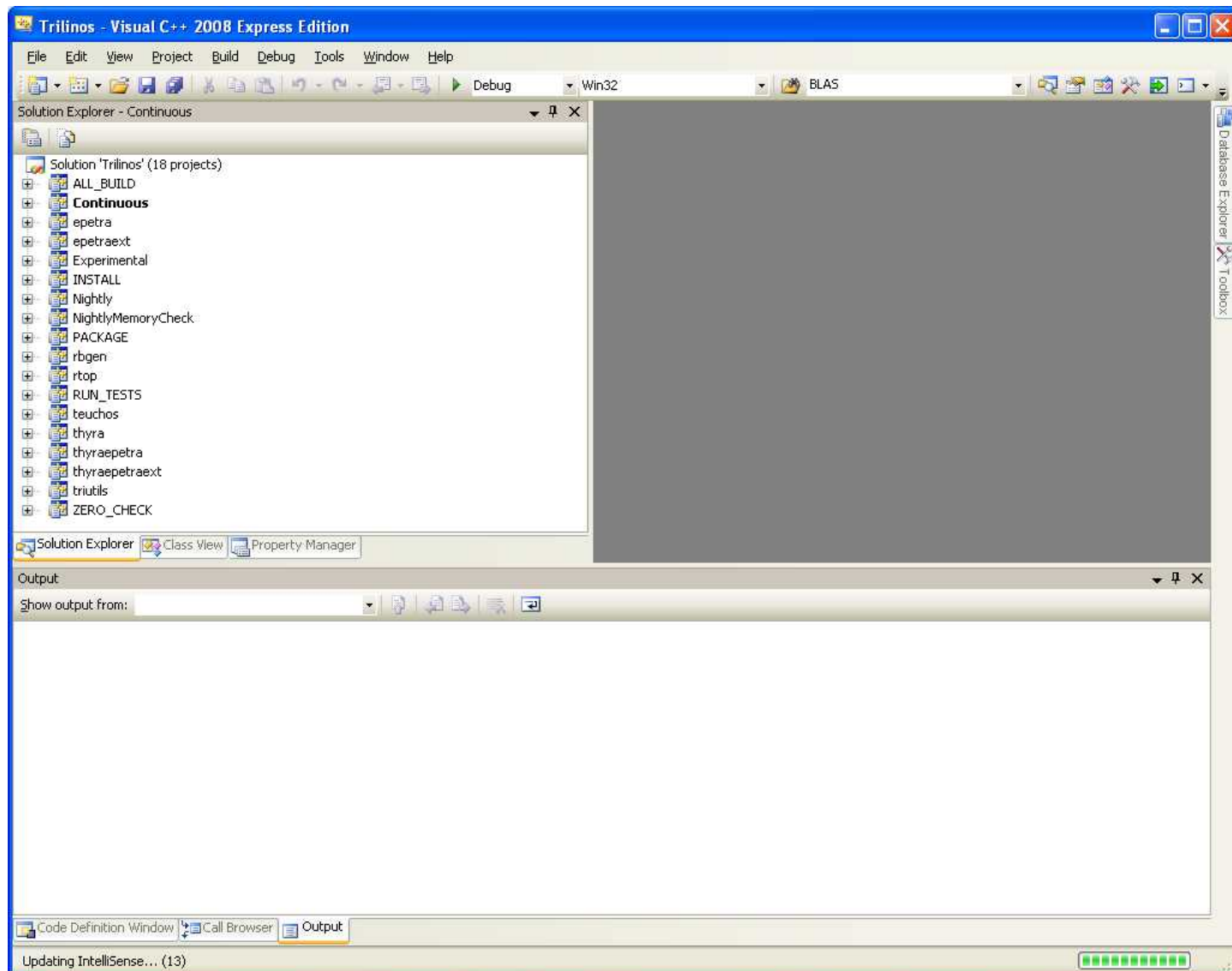
Configuring individual Trilinos packages ...

Configuring done

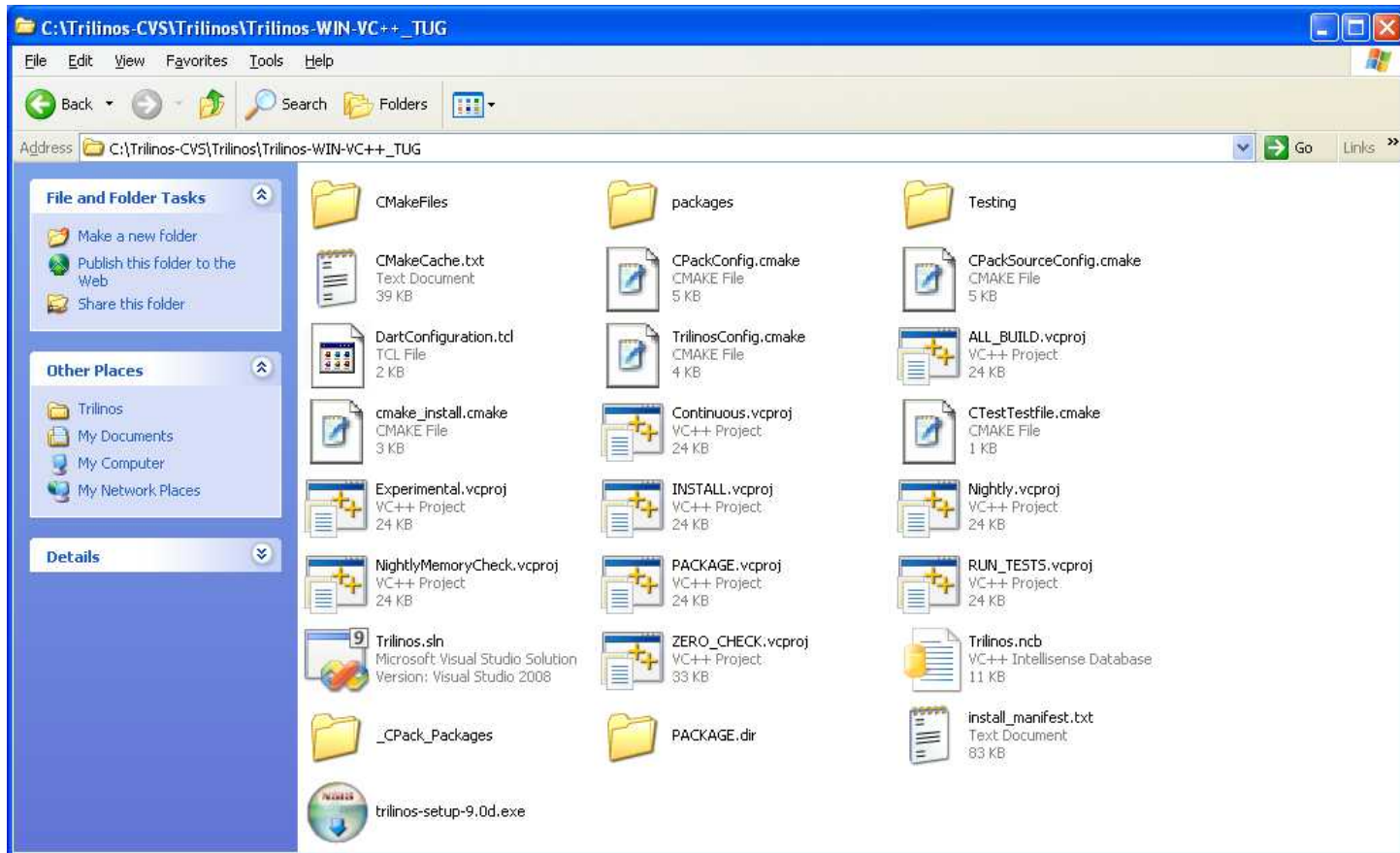
Generated VC++ Project Files



Visual C++ Solution



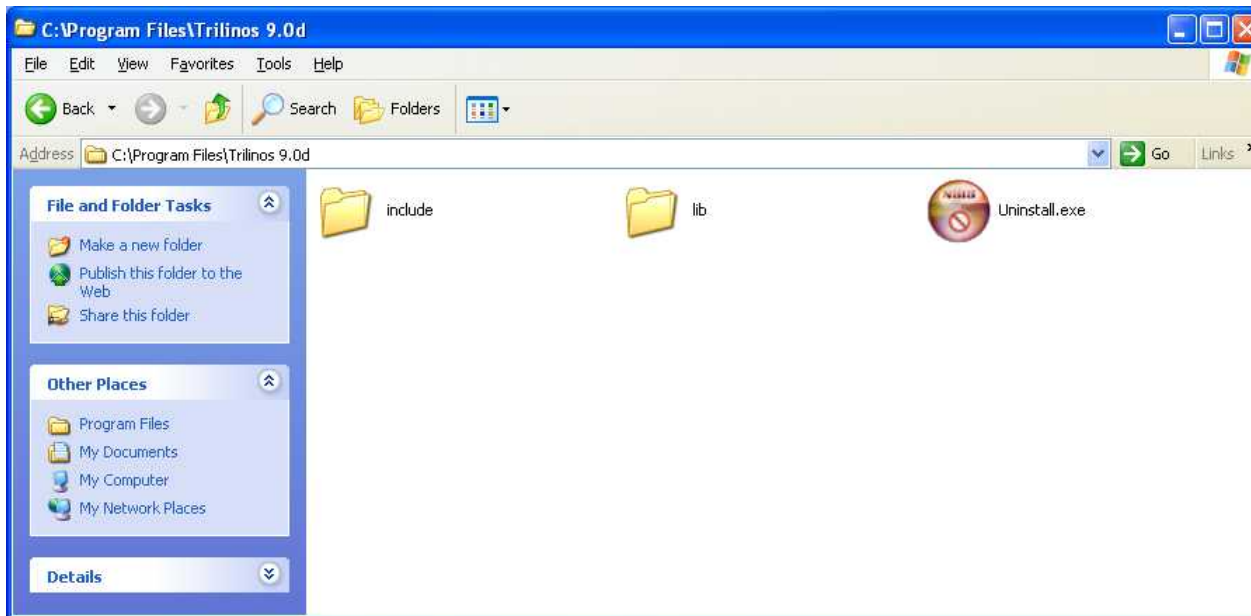
Binary Installer

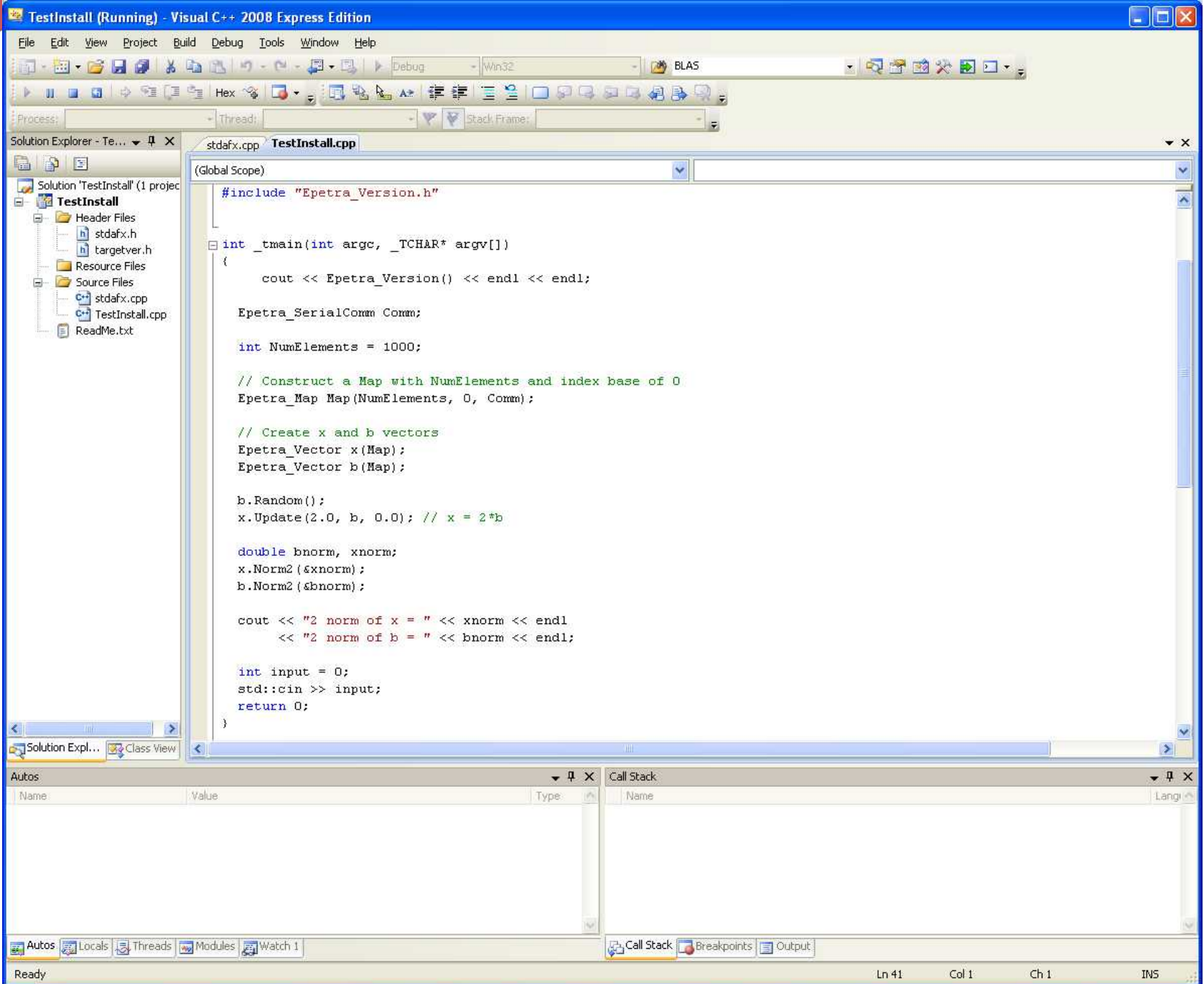


Installing Trilinos

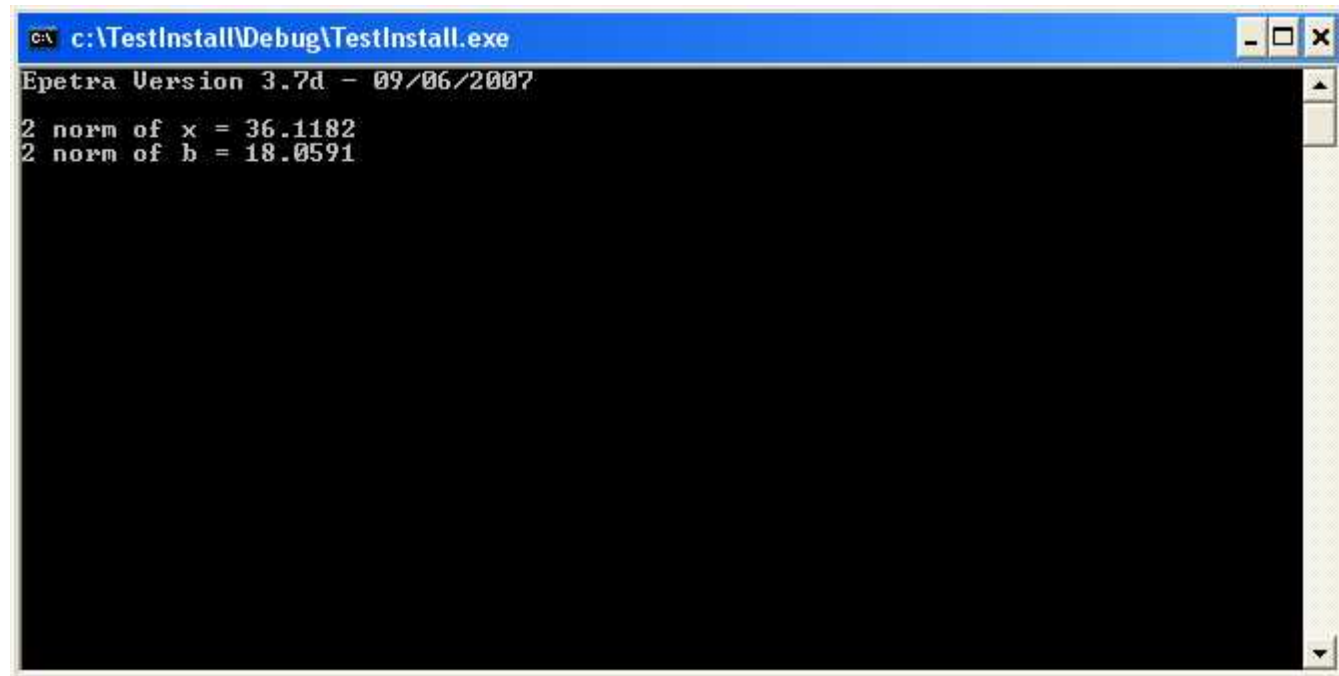


Installed Files





Executing The Epetra Example From VC++



```
c:\TestInstall\Debug\TestInstall.exe
Epetra Version 3.7d - 09/06/2007
2 norm of x = 36.1182
2 norm of b = 18.0591
```



Future for CMake Trilinos?

- Trilinos to start switching over to CMake immediately?
- Provide prototype versions of CMake build system in Trilinos 9.0.x minor releases?
- Options for next major Trilinos release (March 2009?)
 - A) Maintain full Autotools build system and only provide partial support for CMake build system? (Already done)
 - B) Full support for CMake build system for all released Trilinos packages and maintain basic Autotools build system for library install only? (Most likely)
 - C) Full support for CMake build system for all released Trilinos packages and drop Autotools support? (Least likely)

What does the Trilinos user community think about these options?