

Exceptional service in the national interest



Testing and Documenting your Code

Alicia Klinvex

Sandia National Labs

August 8, 2016



Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2016-7497 TR

DOCUMENTATION

Why is documentation important?



- To identify the purpose of the software and its requirements
- To clarify what each component does, what is needed to maintain it, and how it can be reused elsewhere
- To provide user support
 - Minimizes unnecessary handholding of users
- To ensure that software is used within its region of validity
 - Minimizes possibility of producing spurious scientific results

Categories of documentation

- Users guide
- Reference manual
 - List of the interfaces and routines and explanation of functionality
 - Can be generated automatically from code
- Readme files
- Installation guide
- Tutorials

All software needs documentation
Not all software needs a users guide

What is Trilinos?

- A collection of libraries intended to be used as building blocks for the development of scientific applications
- Organized into 66 packages
 - Linear solvers
 - Nonlinear solvers
 - Eigensolvers
 - And more!
- 10,000+ commits
- 135 contributors (according to github)
- Millions of lines of code

How does Trilinos handle documentation?

- Each package does it differently
- User manuals
 - MueLu (algebraic multigrid)
 - AztecOO (Krylov solvers)
 - Teuchos RCP (reference counted pointers)
- Publicly available tutorials, presentations, and slides
 - Tpetra (MPI+X linear algebra)
 - Kokkos
- Well commented examples
- Automatically generated html documentation

Doxxygen

- One approach to producing “reference manual”-like documentation
- Automatically generates html (or LaTeX) documentation from comments in source code
- Easy to update documentation when source code is updated
- Takes Doxyfile as input
- doxywizard - GUI frontend for doxygen
 - Generates Doxyfile for you

Doxxygen demo

- A small linear algebra package: Morpheus
 - Available at <https://github.com/amklinv/morpheus>
- We will document the Vector class
- This linear algebra package will also be used in the software testing hands-on session tonight

TESTING

Outline

- Why testing is important
- Types of tests
- Testing tips
- How Trilinos is tested
- Code coverage
- Verification

Why testing is important: the protein structures of Geoffrey Chang

- Some inherited code flipped two columns of data, inverting an electron-density map
- Resulted in an incorrect protein structure
- Resulted in 5 retracted publications
 - One was cited 364 times
- Many papers and grant applications conflicting with his results were rejected

Why testing is important: the 40 second flight of the Ariane 5

- Ariane 5: a European orbital launch vehicle meant to lift 20 tons into low Earth orbit
- Initial rocket went off course, started to disintegrate, then self-destructed less than a minute after launch
- Seven variables were at risk of leading to an Operand Error (due to conversion of floating point to integer)
 - Four were protected
- Investigation concluded insufficient test coverage as one of the causes for this accident
- Resulted in a loss of \$370,000,000.

Why testing is important: the Therac-25 accidents

- Therac-25: a computer-controlled radiation therapy machine
- Minimal software testing
- Race condition in the code went undetected
- Unlucky patients were struck with approximately 100 times the intended dose of radiation, $\sim 15,000$ rads
- Error code indicated that no dose of radiation was given, so operator instructed machine to proceed
 - Documentation gave no indication that the frequent malfunctions of the machine could place a patient at risk
 - See also: why documentation is important
- Recalled after six accidents resulting in death and serious injuries

Granularity of tests

- Unit tests
 - Test individual functions or classes
 - Build and run fast
 - Localize errors
- Integration tests
 - Test interaction of larger pieces of software
- System-level tests
 - Test the full software system at the user interaction level

Types of tests

- Verification tests
 - Does the code implement the intended algorithm correctly?
 - Check for specific mathematical properties
- Acceptance tests
 - Assert acceptable functioning for a specific customer
 - Generally at the system-level
- Regression (no-change) tests
 - Compare current observable output to a gold standard
 - Must independently verify that the gold standard is correct
- Performance tests
 - Focus on the runtime and resource utilization
 - Nothing to do with correctness
- Installation tests
 - Verify that the configure-make-install is working as expected

CSE testing challenges

- Floating point issues
 - Different results
 - On different platforms
 - On different runs (due to multi-processor computation)
 - Ill-conditioning can magnify these small differences
 - Final solution may be different
 - Number of iterations may be different
 - Performing a diff is bad
- Non-unique solutions

CSE testing challenges

- Scalability testing
 - Difficult to get accurate data on a shared machine
 - Getting access to many processors on a parallel machine is expensive
 - Many supercomputing facilities discourage routine scalability testing
 - Large jobs may sit in the queue for quite some time
 - How do you scale a problem for weak scaling studies?
 - A more refined problem may not have the same condition number

Testing tips

- Ideal time to build a test suite is during development
 - Ensures that new code does not break existing functionality
- Failing tests should help you identify what part of the code needs to be fixed
- Software should be tested regularly
- Develop a consistent policy on dealing with failed tests
 - Use an issue tracking system
 - Add a regression test after the issue is fixed
- Run a regression test suite when checking in new code
- Avoid zero-diffing tests against gold standard output
 - spiff (<https://github.com/dontcallmedom/spiff>)

How is Trilinos tested?

- Trilinos has 1500 tests between its 66 packages
- Developers are strongly advised to run a checkin test script when committing
 - Detects which packages were modified by your commits
 - Determines which packages you potentially broke
 - Configures, builds, and tests those packages
 - On success, pushes to repo
 - On failure, reports why it failed
 - Useful for ensuring your changes don't break another package
 - May take a while, but many people run it overnight
- Automated testing on a variety of different platforms

Why do we do automated testing if everyone uses the checkin script?



- May test a different set of packages
- May test different environments
 - Do your changes work with Intel compilers as well as GNU?
 - Do your changes work on a mac?
 - Do your changes work with CUDA?
- Identifies a small set of commits that could have broken a build or test
 - Average 12 commits per day
 - Identifies the person who knows how to un-break it
- Bugs are easier to fix if caught early

Checkin test script examples

- Example 1: a harmless change to a comment
- Example 2: breaking the build
- Example 3: breaking some tests

Example 1: a harmless change

Applications Places System

anasazi/s : vim

File Edit View Scrollback Bookmarks Settings Help

```
// @HEADER
// ****
//
//      Anasazi: Block Eigensolvers Package
//      Copyright (2004) Sandia Corporation
//
// Under terms of Contract DE-AC04-94AL85000, there is a non-exclusive
// license for use of this work by or on behalf of the U.S. Government.
//
// This library is free software; you can redistribute it and/or modify
// it under the terms of the GNU Lesser General Public License as
// published by the Free Software Foundation; either version 2.1 of the
// License, or (at your option) any later version.
//
// This library is distributed in the hope that it will be useful, but
// WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
// Lesser General Public License for more details.
//
// You should have received a copy of the GNU Lesser General Public
// License along with this library; if not, write to the Free Software
// Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301
// USA
// Questions? Contact Michael A. Heroux (maherou@sandia.gov)
//
// ****
// @HEADER

/*! \file AnasaziTraceMinDavidson.hpp
 * \brief Implementation of the TraceMin-Davidson eigensolver */
#ifndef ANASAZI_TRACEMIN_DAVIDSON_HPP
#define ANASAZI_TRACEMIN_DAVIDSON_HPP

#include "AnasaziConfigDefs.hpp"
#include "AnasaziEigensolver.hpp"
#include "AnasaziMultiVecTraits.hpp"
-- INSERT --

```

anasazi/s : vim

[System Monitor] Fix build of xSDKTrilino... CHECKIN : checkin-tes... anasazi/s : vim USInformationSheet_I... *Unsaved Document 1 ...

Thu Apr 21, 4:09 PM Alicia Klinvex

30,61 Top

Example 1: a harmless change

Applications Places System

CHECKIN : checkin-test-am

Thu Apr 21, 4:42 PM Alicia Klinvex

```
[amklinv@s995692 CHECKIN]$ ./checkin-test-amklinv --do-all
*****
Script: checkin-test.py \
--src-dir='/home/amklinv/TrilinosDir/github/Trilinos' \
--default-builds='MPI_DEBUG,SERIAL_RELEASE' \
--extra-repos-file='' \
--extra-repos-type='' \
--extra-repos='' \
--require-extra-repos-exist \
--enable-packages='' \
--disable-packages='FEI,Moertel,STK,Phalanx,PyTrilinos' \
--enable-all-packages='auto' \
--enable-fwd-packages \
--continue-if-no-updates \
--continue-if-no-changes-to-push \
--continue-if-no-enables \
--extra-cmake-options='' \
--test-categories='BASIC' \
--make-options='-j48' \
--ctest-options='-E '(Anasazi_Epetra_BlockDavidson_test|Anasazi_Epetra_BlockDavidson_auxtest_MPI_4|Anasazi_Epetra_GeneralizedDavidson_solverstest_MPI_4|Anasazi_BlockDavidsonThyra_test_MPI_4|NOX_Thyra_Heq_MPI_1|Rythmos_BackwardEuler_ConvergenceTest_MPI_1|Rythmos_IntegratorBuilder_ConvergenceTest_MPI_1|NOX_Thyra_Heq)' \
--ctest-timeout=60.0 \
--no-show-all-tests \
--st-extra-builds='' \
--extra-builds='' \
--send-email-to='amklinv@sandia.gov' \
--skip-case-send-email \
--send-build-case-email=always \
--send-email-for-all \
--send-email-to-on-push='trilinos-checkin-tests@software.sandia.gov' \
--no-force-push \
--do-push-readiness-check \
--rebase \
--append-test-results \
--do-all \

```

Starting time: Thu Apr 21 16:11:38 MDT 2016

...IN : checkin-test-am

[System Monitor] Klinvex, Alicia Marie - ... CHECKIN : checkin-test-am MPI_DEBUG : bash USInformationSheet_I... *Unsaved Document 1 ...

Example 1: a harmless change

Applications Places System 

CHECKIN : checkin-test-am  Thu Apr 21, 4:44 PM Alicia Klinvex

```

File Edit View Scrollback Bookmarks Settings Help
Determining the set of packages to enable by examining /home/amklinv/TrilinosDir/github/Trilinos/CHECKIN/modifiedFiles.out ...

Modified file: 'packages/anasazi/src/AnasaziTraceMinDavidson.hpp'
  => Enabling 'Anasazi'!

Full package enable list: [Anasazi]

Removing package enables: [FEI,Moertel,STK,Phalanx,PyTrilinos]

Filtering the set of enabled packages according to allowed package types ...

Final package enable list: [Anasazi]

Enabling forward packages on request!

Adding hard disables for specified packages 'FEI,Moertel,STK,Phalanx,PyTrilinos' ...

cmakePkgOptions: [u'-DTrilinos_ENABLE_Anasazi:BOOL=ON', '-DTrilinos_ENABLE_ALL_OPTIONAL_PACKAGES:BOOL=ON', '-DTrilinos_ENABLE_ALL_FORWARD_DEP_PACKAGES:BOOL=ON',
  '-DTrilinos_ENABLE_FEI:BOOL=OFF', '-DTrilinos_ENABLE_Moertel:BOOL=OFF', '-DTrilinos_ENABLE_STK:BOOL=OFF', '-DTrilinos_ENABLE_Phalanx:BOOL=OFF', '-DTrilinos_ENABLE_PyTrilinos:BOOL=OFF']

cmakeOptions = ['-DTrilinos_TRIBITS_DIR:PATH=/home/amklinv/TrilinosDir/github/Trilinos/cmake/tribits', '-DTrilinos_ENABLE_TESTS:BOOL=ON', '-DTrilinos_TEST_CATEGORIES:STRING=BASIC',
  '-DTrilinos_ALLOW_NO_PACKAGES:BOOL=OFF', '-DDART_TESTING_TIMEOUT:STRING=60.0', '-DTPL_ENABLE_Pthread:BOOL=OFF', '-DTPL_ENABLE_BinUtils:BOOL=OFF',
  '-DTrilinos_ENABLE_SECONDARY_TESTED_CODE:BOOL=OFF', '-DTPL_ENABLE_MPI:BOOL=ON', '-DCMAKE_BUILD_TYPE:STRING=RELEASE', '-DTrilinos_ENABLE_DEBUG:BOOL=ON',
  '-DTrilinos_ENABLE_CHECKED_STL:BOOL=ON', '-DTrilinos_ENABLE_DEBUG_SYMBOLS:BOOL=ON', '-DTrilinos_ENABLE_EXPLICIT_INSTANTIATION:BOOL=ON', '-DTeuchos_ENABLE_DEFAULT_STACKTRACE:BOOL=OFF',
  '-D Trilinos_ENABLE_EXPLICIT_INSTANTIATION:BOOL=ON', '-D CMAKE_INSTALL_PREFIX:PATH="/home/amklinv/TrilinosDir/trilinos-install"',
  '-D CMAKE_CXX_FLAGS:STRING="-Wall -ansi -pedantic -Wshadow"', '-D MPI_EXEC_MAX_NUMPROCS:STRING=8', '-D HAVE_GCC_ABI_DEMANGLE:BOOL=ON', '-D Trilinos_ENABLE_OpenMP:BOOL=OFF',
  '-D TPL_ENABLE_QT:BOOL=OFF', '-D BLAS_LIBRARY_NAMES:STRING="libf77blas.so.3"', '-D BLAS_LIBRARY_DIRS:PATH="/usr/lib64/atlas", "-D LAPACK_LIBRARY_NAMES:STRING="liblapack.so.3"',
  '-D LAPACK_LIBRARY_DIRS:PATH="/usr/lib64/atlas"', '-D TPL_ENABLE_Netcdf:OFF', '-D CMAKE_BUILD_TYPE:STRING=DEBUG', '-D Teuchos_ENABLE_DEBUG:BOOL=ON',
  '-D Kokkos_ENABLE_BOUNDS_CHECK:BOOL=ON', '-D Kokkos_ENABLE_DEBUG:BOOL=ON', '-D TPL_ENABLE_MPI:BOOL=ON', '-DTrilinos_ENABLE_Anasazi:BOOL=ON',
  '-DTrilinos_ENABLE_ALL_OPTIONAL_PACKAGES:BOOL=ON', '-DTrilinos_ENABLE_ALL_FORWARD_DEP_PACKAGES:BOOL=ON', '-DTrilinos_ENABLE_FEI:BOOL=OFF', '-DTrilinos_ENABLE_Moertel:BOOL=OFF',
  '-DTrilinos_ENABLE_STK:BOOL=OFF', '-DTrilinos_ENABLE_Phalanx:BOOL=OFF', '-DTrilinos_ENABLE_PyTrilinos:BOOL=OFF']

Creating base configure file do-configure.base ...

Running: chmod a+x do-configure.base

```

Note that the checkin script correctly identified what was modified.

Example 1: a harmless change

Applications Places System

CHECKIN : checkin-test-am

Thu Apr 21, 4:46 PM Alicia Klinvex

```

File Edit View Scrollback Bookmarks Settings Help
E) Analyze the overall results and send email notification (MPI_DEBUG) ...

E.1) Determine what passed and failed ...

The pull passed!

The configure passed!

The build passed!

testResultsLine = '100% tests passed, 0 tests failed out of 237'
All of the tests ran passed!

E.2) Construct the email message ...

subjectLine = 'passed: Trilinos/MPI_DEBUG: passed=237,notpassed=0'

Running: touch email.success

E.3) Send the email message ...

Running: mailx -s "passed: Trilinos/MPI_DEBUG: passed=237,notpassed=0" amklinv@sandia.gov < email.out

*** Doing build and test of SERIAL_RELEASE ...
***
```

...IN : checkin-test-am

[System Monitor] Klinvex, Alicia Marie ... CHECKIN : checkin-test-am MPI_DEBUG : bash USInformationSheet_... *Unsaved Document... snapshot4.png snapshot5.png

Configure, build, and test
passed for MPI_DEBUG

Example 1: a harmless change

Applications Places System

CHECKIN : checkin-test-am

Thu Apr 21, 4:47 PM Alicia Klinvex

```

READY TO PUSH: Trilinos: s995692.srn.sandia.gov
Thu Apr 21 16:22:57 MDT 2016

Enabled Packages: Anasazi
Disabled Packages: FEI,Moertel,STK,Phalanx,PyTrilinos
Enabled all Forward Packages

Build test results:
-----
0) MPI_DEBUG => passed: passed=237,notpassed=0 (8.42 min)
1) SERIAL_RELEASE => passed: passed=243,notpassed=0 (2.71 min)

*** Commits for repo :
 982db3b Anasazi: Modified a comment in TraceMin-Davidson

0) MPI_DEBUG Results:
-----
passed: Trilinos/MPI_DEBUG: passed=237,notpassed=0

Thu Apr 21 16:20:14 MDT 2016

Enabled Packages: Anasazi
Disabled Packages: FEI,Moertel,STK,Phalanx,PyTrilinos
Enabled all Forward Packages
Hostname: s995692.srn.sandia.gov
Source Dir: /home/amklinv/TrilinosDir/github/Trilinos
Build Dir: /home/amklinv/TrilinosDir/github/Trilinos/CHECKIN/MPI_DEBUG

CMake Cache Variables: -DTrilinos_TRIBITS_DIR:/home/amklinv/TrilinosDir/github/Trilinos/cmake/tribits -DTrilinos_ENABLE_TESTS:BOOL=ON -DTrilinos_TEST_CATEGORIES:STRING=BASIC -DTrilinos_ALLOW_NO_PACKAGES:BOOL=OFF -DDART_TESTING_TIMEOUT:STRING=60.0 -DTPL_ENABLE_Pthread:BOOL=OFF -DTPL_ENABLE_BinUtils:BOOL=OFF -DTrilinos_ENABLE_SECONDARY_TESTED_CODE:BOOL=OFF -DTPL_ENABLE_MPI:BOOL=ON -DCMAKE_BUILD_TYPE:STRING=RELEASE -DTrilinos_ENABLE_DEBUG:BOOL=ON -DTrilinos_ENABLE_CHECKED_STL:BOOL=ON -DTrilinos_ENABLE_DEBUG_SYMBOLS:BOOL=ON -DTrilinos_ENABLE_EXPLICIT_INSTANTIATION:BOOL=ON -DTeuchos_ENABLE_DEFAULT_STACKTRACE:BOOL=OFF -DTrilinos_ENABLE_EXPLICIT_INSTANTIATION:BOOL=ON -D CMAKE_INSTALL_PREFIX:PATH="/home/amklinv/TrilinosDir/trilinos-install" -D CMAKE_CXX_FLAGS:STRING="-Wall -ansi -pedantic -Wshadow" -D MPI_EXEC_MAX_NUMPROCS:STRING=8 -D HAVE_GCC_ABI_DEMANGLE:BOOL=ON -D Trilinos_ENABLE_OpenMP:BOOL=OFF -D TPL_ENABLE_QT:BOOL=OFF -D BLAS_LIBRARY_NAMES:STRING="libf77blas.so.3" -D BLAS_LIBRARY_DIRS:PATH="/usr/lib64/atlas" -D LAPACK_LIBRARY_NAMES:STRING="liblapack.so.3" -D LAPACK_LIBRARY_DIRS:PATH="/usr/lib64/atlas" -D TPL_ENABLE_Netcdf=OFF -D CMAKE_BUILD_TYPE:STRING=DEBUG -D Teuchos_ENABLE_DEBUG:BOOL=ON -D Kokkos_ENABLE_BOUNDS_CHECK:BOOL=ON -D Kokkos_ENABLE_DEBUG:BOOL=ON -D TPL_ENABLE_MPI:BOOL=ON -DTrilinos_ENABLE_Anasazi:BOOL=ON -DTrilinos_ENABLE_ALL_OPTIONAL_PACKAGES:BOOL=ON -DTrilinos_ENABLE_ALL_FORWARD_D
```

We are ready to push because all tests passed

...IN : checkin-test-am

[System Monitor] [Klinvex, Alicia Marie ...] [CHECKIN : checkin-test-am] [MPI_DEBUG : bash] [USInformationSheet ...] [*Unsaved Document... snapshot4.png snapshot5.png]

Example 2: broken build

Applications Places System Thu Apr 21, 4:50 PM Alicia Klinvex

File Edit View Scrollback Bookmarks Settings Help

```

const Teuchos::RCP<StatusTest<ScalarType,MV,OP> >      &tester,
const Teuchos::RCP<MatOrthoManager<ScalarType,MV,OP> > &ortho,
Teuchos::ParameterList &params
);

private:
//  

// Convenience typedefs  

//  

typedef MultiVecTraits<ScalarType,MV> MVT;  

typedef OperatorTraits<ScalarType,MV,OP> OPT;  

typedef Teuchos::ScalarTraits<ScalarType> SCT;  

typedef typename SCT::magnitudeType MagnitudeType;

// TraceMin specific methods
void addToBasis(const Teuchos::RCP<const MV> Delta);

void harmonicAddToBasis(const Teuchos::RCP<const MV> Delta);
} ←
// Implementations
//  

// Constructors
template <class ScalarType, class MV, class OP>
TraceMinDavidson<ScalarType,MV,OP>::TraceMinDavidson(
    const Teuchos::RCP<Eigenproblem<ScalarType,MV,OP> >      &problem,
    const Teuchos::RCP<SortManager<typename Teuchos::ScalarTraits<ScalarType>::magnitudeType> > &sorter,
    const Teuchos::RCP<OutputManager<ScalarType> >           &printer,
    const Teuchos::RCP<StatusTest<ScalarType,MV,OP> >      &tester,
-- INSERT --

```

Missing semicolon at the end of the class. This will break the build

anasazi/s : vim 99, 4 28%

[System Monitor] Fix build of xSDKTrilino... CHECKIN : checkin-tes... anasazi/s : vim USInformationSheet_I... *Unsaved Document 1 ...

Example 2: broken build

```

Applications Places System
File Edit View Scrollback Bookmarks Settings Help
C) Do the build (MPI_DEBUG) ...

Running: make -j48
Writing console output to file make.out ...
Runtime for command = 0.235778 minutes
Build failed returning 2!

Traceback (most recent call last):
  File "/home/amklinv/TrilinosDir/github/Trilinos/cmake/tribits/ci_support/CheckinTest.py", line 1586, in runBuildTestCase
    raise Exception("Build failed!")
Exception: Build failed!
E) Analyze the overall results and send email notification (MPI_DEBUG) ...

E.1) Determine what passed and failed ...

The pull passed!

The configure passed!

The build FAILED! ←
The tests were never even run!

E.2) Construct the email message ...

subjectLine = 'FAILED: Trilinos/MPI_DEBUG: build failed'

...IN : checkin-test-am
[System Monitor] Klinvex, Alicia Marie - ... CHECKIN : checkin-test... anasazi/s : bash USinformationSheet_I... *Unsaved Document 1 ...
  
```

The checkin script detected that I broke the build

Example 2: broken build

Applications Places System  Thu Apr 21, 4:56 PM Alicia Klinvex

MPI_DEBUG : bash

```

File Edit View Scrollback Bookmarks Settings Help
Built target Rythmos_StepperHelpers_UnitTest
[100%] [100%] Built target Rythmos_Thyra_UnitTest
Built target Rythmos_TimeRange_UnitTest
[100%] [100%] [100%] Built target Rythmos_ForwardEuler_ConvergenceTest
[100%] Built target Rythmos_VanderPolModel_UnitTest
Built target Rythmos_BackwardEuler_ConvergenceTest
[100%] Built target Rythmos_ExplicitRK_ConvergenceTest
[100%] Built target Rythmos_ImplicitBDF_ConvergenceTest
Built target Rythmos_IntegratorBuilder_ConvergenceTest
[100%] Built target Rythmos_simpleAdjoint
In file included from /home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/src/AnasaziTraceMinDavidsonSolMgr.hpp:40:0,
                 from /home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/tpetra/test/TraceMinDavidson/cxx_main_standard_noprec.cpp:8:
/home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/src/AnasaziTraceMinDavidson.hpp:99:3: error: expected ';' after class definition
In file included from /home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/src/AnasaziTraceMinDavidsonSolMgr.hpp:40:0,
                 from /home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/tpetra/example/TraceMinDavidson/TraceMinDavidsonUserOpEx.cpp:8:
/home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/src/AnasaziTraceMinDavidson.hpp:99:3: error: expected ';' after class definition
In file included from /home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/src/AnasaziTraceMinDavidsonSolMgr.hpp:40:0,
                 from /home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/tpetra/example/TraceMinDavidson/TraceMinDavidsonLaplacianEx.cpp:8:
/home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/src/AnasaziTraceMinDavidson.hpp:99:3: error: expected ';' after class definition
In file included from /home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/src/AnasaziTraceMinDavidsonSolMgr.hpp:40:0,
                 from /home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/tpetra/example/TraceMinDavidson/TraceMinDavidsonGeneralizedEx.cpp:8:
/home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/src/AnasaziTraceMinDavidson.hpp:99:3: error: expected ';' after class definition
In file included from /home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/src/AnasaziTraceMinDavidsonSolMgr.hpp:40:0,
                 from /home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/tpetra/example/TraceMinDavidson/TraceMinDavidsonSpecTransEx.cpp:8:
/home/amklinv/TrilinosDir/github/Trilinos/packages/anasazi/src/AnasaziTraceMinDavidson.hpp:99:3: error: expected ';' after class definition
make[2]: *** [packages/anasazi/tpetra/test/TraceMinDavidson/CMakeFiles/Anasazi_Tpetra_TraceMinDavidson_largest_standard_test.dir/cxx_main_standard_noprec.cpp.o] Error 1
make[1]: *** [packages/anasazi/tpetra/test/TraceMinDavidson/CMakeFiles/Anasazi_Tpetra_TraceMinDavidson_largest_standard_test.dir/all] Error 2
make[1]: *** Waiting for unfinished jobs...
make[2]: *** [packages/anasazi/tpetra/example/TraceMinDavidson/CMakeFiles/Anasazi_Tpetra_TraceMinDavidsonUserOpEx.cpp.o] Error 1
make[1]: *** [packages/anasazi/tpetra/example/TraceMinDavidson/CMakeFiles/Anasazi_Tpetra_TD_UserOp_example.dir/TraceMinDavidsonUserOpEx.cpp.o] Error 2
make[2]: *** [packages/anasazi/tpetra/example/TraceMinDavidson/CMakeFiles/Anasazi_Tpetra_TD_Gen_example.dir/TraceMinDavidsonGeneralizedEx.cpp.o] Error 1
make[1]: *** [packages/anasazi/tpetra/example/TraceMinDavidson/CMakeFiles/Anasazi_Tpetra_TD_Gen_example.dir/all] Error 2
make[2]: *** [packages/anasazi/tpetra/example/TraceMinDavidson/CMakeFiles/Anasazi_Tpetra_TD_Fiedler_example.dir/TraceMinDavidsonLaplacianEx.cpp.o] Error 1
make[1]: *** [packages/anasazi/tpetra/example/TraceMinDavidson/CMakeFiles/Anasazi_Tpetra_TD_Fiedler_example.dir/all] Error 2
make[2]: *** [packages/anasazi/tpetra/example/TraceMinDavidson/CMakeFiles/Anasazi_Tpetra_TD_Trans_example.dir/TraceMinDavidsonSpecTransEx.cpp.o] Error 1
make[1]: *** [packages/anasazi/tpetra/example/TraceMinDavidson/CMakeFiles/Anasazi_Tpetra_TD_Trans_example.dir/all] Error 2
make: *** [all] Error 2
[amklinv@s995692 MPI_DEBUG]$ 
```

MPI_DEBUG : bash

[System Monitor] Klinvex, Alicia Marie - ... MPI_DEBUG : bash anasazi/s : bash USInformationSheet_I... *Unsaved Document 1 ...

Checkin script also creates
a log file with the error

Example 3: broken tests

Applications Places System  Thu Apr 21, 5:02 PM Alicia Klinvex

File Edit View Scrollback Bookmarks Settings Help

```

useMultipleShifts_ = params.get("Use Multiple Shifts", true);

shiftThresh_ = params.get("Shift Threshold", 1e-2);
useRelShiftThresh_ = params.get("Relative Shift Threshold", true);
shiftNorm_ = params.get("Shift Norm", "2");
TEUCHOS_TEST_FOR_EXCEPTION(shiftNorm_ != "2" && shiftNorm_ != "M", std::invalid_argument,
    "Anasazi::TraceMin::constructor: Invalid value for \"Shift Norm\"; valid options are \"2\", \"M\".");
considerClusters_ = params.get("Consider Clusters", true);

projectAllVecs_ = params.get("Project All Vectors", true);
projectLockedVecs_ = params.get("Project Locked Vectors", true);
useRHSR_ = params.get("Use Residual as RHS", true);
useHarmonic_ = params.get("Use Harmonic Ritz Values", false);
computeAllRes_ = params.get("Compute All Residuals", true);

// set the block size and allocate data
int bs = params.get("Block Size", problem_->getNEV());
int nb = params.get("Num Blocks", 1);
// setSize(bs,nb); ←
NEV_ = problem_->getNEV();

// Create the Ritz shift operator
ritzOp_ = rcp (new tracemin_ritz_op_type (Op_, MOp_, Prec_));

// Set the maximum number of inner iterations
const int innerMaxIts = params.get ("Maximum Krylov Iterations", 200);
ritzOp_->setMaxIts (innerMaxIts);

alpha_ = params.get ("HSS: alpha", ONE);
}

///////////////////////////////
// Destructor
template <class ScalarType, class MV, class OP>
TraceMinBase<ScalarType,MV,OP>::~TraceMinBase() {}

-- INSERT --

```

anasazi/s : vim

[System Monitor] Klinvex, Alicia Marie - ... anasazi/s : vim modifyAnasazi : bash USInformationSheet_I... *Unsaved Document 1 ...

746,3 19%

Added a logic error to the code.

Example 3: broken tests

Applications Places System

CHECKIN : checkin-test-am

Thu Apr 21, 5:16 PM Alicia Klinvex

File Edit View Scrollback Bookmarks Settings Help

FAILED CONFIGURE/BUILD/TEST: Trilinos: s995692.srn.sandia.gov

Thu Apr 21 17:14:53 MDT 2016

Enabled Packages: Anasazi
 Disabled Packages: FEI,Moertel,STK,Phalanx,PyTrilinos
 Enabled all Forward Packages

Build test results:

```
0) MPI_DEBUG => FAILED: passed=233,notpassed=4 => Not ready to push! (8.43 min)
1) SERIAL_RELEASE => FAILED: passed=239,notpassed=4 => Not ready to push! (2.74 min)
```

Failed because one of the build/test cases failed!

*** Commits for repo :
 6bb949b Anasazi: Broke some TraceMin tests. Oops!

0) MPI_DEBUG Results:

```
FAILED: Trilinos/MPI_DEBUG: passed=233,notpassed=4
```

Thu Apr 21 17:12:09 MDT 2016

Enabled Packages: Anasazi
 Disabled Packages: FEI,Moertel,STK,Phalanx,PyTrilinos
 Enabled all Forward Packages
 Hostname: s995692.srn.sandia.gov
 Source Dir: /home/amklinv/TrilinosDir/github/Trilinos
 Build Dir: /home/amklinv/TrilinosDir/github/Trilinos/CHECKIN/MPI_DEBUG

CMake Cache Variables: -DTrilinos_TRIBITS_DIR=/home/amklinv/TrilinosDir/github/Trilinos/cmake/tribits -DTrilinos_ENABLE_TESTS:BOOL=ON -DTrilinos_TEST_CATEGORIES:STRING=BASIC -DTrilinos_ALLOW_NO_PACKAGES:BOOL=OFF -DDART_TESTING_TIMEOUT:STRING=60.0 -DTPL_ENABLE_Pthread:BOOL=OFF -DTPL_ENABLE_BinUtils:BOOL=OFF -DTrilinos_ENABLE_SECONDARY_TESTED_CODE:BOOL=OFF -DTPL_ENABLE_MPI:BOOL=ON -DCMAKE_BUILD_TYPE:STRING=RELEASE -DTrilinos_ENABLE_DEBUG:BOOL=ON -DTrilinos_ENABLE_CHECKED_STL:BOOL=ON -DTrilinos_ENABLE_DEBUG_SYMBOLS:BOOL=ON -DTrilinos_ENABLE_EXPLICIT_INSTANTIATION:BOOL=ON -DTeuchos_ENABLE_DEFAULT_STACKTRACE:BOOL=OFF -D Trilinos_ENABLE_EXPLICIT_INSTANTIATION:BOOL=ON -D CMAKE_INSTALL_PREFIX:PATH="/home/amklinv/TrilinosDir/trilinos-install" -D CMAKE_CXX_FLAGS:STRING="-Wall -ansi -pedantic -Wshadow" -D MPI_EXEC_MAX_NUMPROCS:STRING=8 -D HAVE_GCC_ABI_DEMANGLE:BOOL=ON -D Trilinos_ENABLE_OpenMP:BOOL=OFF -D TPL_ENABLE_QT:BOOL=OFF -D BLAS_LIB

...IN : checkin-test-am

[System Monitor] Klinvex, Alicia Marie - ... CHECKIN : checkin-test-am MPI_DEBUG : bash USInformationSheet_I... *Unsaved Document 1 ... modifyAnasazi : bash

The checkin script detected
that I broke several tests

Example 3: broken tests

Applications Places System

MPI_DEBUG : bash

File Edit View Scrollback Bookmarks Settings Help

```

Start 227: Rythmos_complicatedExample_amesos_nox_bdf_MPI_1 ..... Passed 3.52 sec
227/237 Test #227: Rythmos_complicatedExample_amesos_nox_bdf_MPI_1 ..... Passed 0.39 sec
Start 228: Rythmos_complicatedExample_amesos_nox_bd_MPI_1 ..... Passed 0.45 sec
228/237 Test #228: Rythmos_complicatedExample_amesos_nox_bd_MPI_1 ..... Passed 0.47 sec
Start 229: Rythmos_complicatedExample_aztecoo_0_MPI_1 ..... Passed 0.45 sec
229/237 Test #229: Rythmos_complicatedExample_aztecoo_0_MPI_1 ..... Passed 0.47 sec
Start 230: Rythmos_complicatedExample_aztecoo_1_MPI_1 ..... Passed 0.47 sec
230/237 Test #230: Rythmos_complicatedExample_aztecoo_1_MPI_1 ..... Passed 0.47 sec
Start 231: Rythmos_complicatedExample_aztecoo_2_MPI_1 ..... Passed 0.47 sec
231/237 Test #231: Rythmos_complicatedExample_aztecoo_2_MPI_1 ..... Passed 0.47 sec
Start 232: Rythmos_complicatedExample_aztecoo_nox_MPI_1 ..... Passed 0.47 sec
232/237 Test #232: Rythmos_complicatedExample_aztecoo_nox_MPI_1 ..... Passed 0.47 sec
Start 233: Rythmos_complicatedExample_belos_0_MPI_1 ..... Passed 0.47 sec
233/237 Test #233: Rythmos_complicatedExample_belos_0_MPI_1 ..... Passed 0.47 sec
Start 234: Rythmos_complicatedExample_belos_1_MPI_1 ..... Passed 0.47 sec
234/237 Test #234: Rythmos_complicatedExample_belos_1_MPI_1 ..... Passed 0.47 sec
Start 235: Rythmos_complicatedExample_belos_nox_bdf_MPI_1 ..... Passed 0.47 sec
235/237 Test #235: Rythmos_complicatedExample_belos_nox_bdf_MPI_1 ..... Passed 0.47 sec
Start 236: Rythmos_simpleAdjoint_amesos_0_MPI_1 ..... Passed 0.47 sec
236/237 Test #236: Rythmos_simpleAdjoint_amesos_0_MPI_1 ..... Passed 0.47 sec
Start 237: Rythmos_simpleAdjoint_amesos_1_MPI_1 ..... Passed 0.47 sec
237/237 Test #237: Rythmos_simpleAdjoint_amesos_1_MPI_1 ..... Passed 0.47 sec

98% tests passed, 4 tests failed out of 237

Label Time Summary:
Anasazi      = 100.15 sec
NOX          = 165.35 sec
Rythmos      = 124.19 sec

Total Test time (real) = 389.89 sec

The following tests FAILED:
  56 - Anasazi_Tpetra_TraceMin_smallest_proj_test_MPI_4 (Failed)
  57 - Anasazi_Tpetra_TraceMin_smallest_schur_test_MPI_4 (Failed)
  58 - Anasazi_Tpetra_TraceMin_largest_standard_test_MPI_4 (Failed)
  59 - Anasazi_Tpetra_TraceMinDavidson_largest_standard_test_MPI_4 (Failed)

Errors while running CTest
[amklinv@s995692 MPI_DEBUG]$ 
```

The log file tells us which tests were broken

[System Monitor] Klinvex, Alicia Marie - ... CHECKIN : checkin-tes... MPI_DEBUG : bash USInformationSheet_I... *Unsaved Document 1 ... modifyAnasazi : bash

Trilinos automated testing

Login All Dashboards Monday, June 06 2016 08:58:08 MDT

Trilinos Dashboard Calendar Previous Current Project

Project									
Project	Configure			Build			Test		
	Error	Warning	Pass	Error	Warning	Pass	Not Run	Fail	Pass
Trilinos 	1	531	530	0	272	257	0	14	3976
SubProjects									
Project	Configure			Build			Test		
	Error	Warning	Pass	Error	Warning	Pass	Not Run	Fail	Pass
Teuchos	0	21	21	0	12	9	0	0	227
ThreadPool	0	1	1	0	0	1			
Sacado	0	2	2	0	2	0	0	0	564
RTOp	0	20	20	0	0	20			
Kokkos	0	19	19	0	0	19	0	0	9
Epeta	0	21	21	0	12	9	0	1	244
Zoltan	0	21	21	0	13	8	0	0	135
Shards	0	1	1	0	0	1			
GlobiPack	0	1	1	0	0	1			

Trilinos automated testing

Nightly											
Site	Build Name	Update		Configure		Build		Test			
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	Build Time	Labels
artemis.srn.sandia.gov	Linux-intel-15.0.2-MPI_RELEASE_DEV_DownStream_ETI_SERIAL-OFF_OPENMP-ON_PTHREAD-OFF_CUDA-OFF_COMPLEX-OFF	68	1	140	0	216	0	3	1256	6 hours ago	(44 labels)
lightsaber.srn.sandia.gov	Linux-GCC-4.7.2-RELEASE_DEV_MueLu_Matlab	69	0	111	0	51	0	0	431	10 hours ago	(25 labels)
enigma.sandia.gov	Linux-GCC-4.8.3-OPENMPI_1.6.4_DEBUG_DEV_MueLu_Basker	69	0	227	0	117	0	0	96	9 hours ago	(25 labels)
hansel.sandia.gov	Linux-GCC-4.4.7-MPI_OPT_DEV_XYCE	121	0	70	0	28	0	0	553	9 hours ago	(13 labels)
enigma.sandia.gov	Linux-GCC-4.8.3-OPENMPI_1.6.4_DEBUG_DEV_MueLu_KLU2	69	0	225	0	91	0	0	73	8 hours ago	(25 labels)
enigma.sandia.gov	Linux-GCC-4.8.3-OPENMPI_1.6.4_DEBUG_DEV_MueLu_ExtraTypes_EI	69	0	227	0	117	0	0	97	8 hours ago	(25 labels)
enigma.sandia.gov	Linux-GCC-4.8.3-SERIAL_DEBUG_DEV_MueLu_ExtraTypes	69	0	227	0	117	0	3	94	7 hours ago	(25 labels)
enigma.sandia.gov	Linux-GCC-4.8.3-SERIAL_RELEASE_DEV_MueLu_Experimental	69	0	227	0	113	0	4	107	6 hours ago	(25 labels)

Trilinos automated testing

- Several Amesos2 (direct solver) tests are broken.

SubProject Dependencies										
Project	Configure			Build			Test			Last submission
	Error	Warning	Pass	Error	Warning	Pass	Not Run	Fail	Pass	
Teuchos	0	22	22	0	13	9	0	0	227	2016-06-06 09:01:20
Epetra	0	22	22	0	13	9	0	1	244	2016-06-06 09:02:05
Triutils	0	22	22	0	0	21	0	0	2	2016-06-06 09:02:16
Tpetra	0	20	20	0	18	2	0	0	285	2016-06-06 08:10:13
EpetraExt	0	21	21	0	3	18	0	0	26	2016-06-06 08:11:16
ThreadPool	0	1	1	0	0	1				2016-06-06 02:51:44
Amesos	0	21	21	0	1	20	0	0	41	2016-06-06 08:16:59

- Are any of its dependencies broken?
 - Yes, there is a broken Epetra (basic linear algebra) test
 - Maybe this broke Amesos2

Trilinos automated testing

- Which tests were broken in Amesos2?

Testing started on 2016-06-06 07:42:35

Site Name:enigma.sandia.gov

Build Name:Linux-GCC-4.8.3-SERIAL_DEBUG_DEV_MueLu_ExtraTypes

Total time:16s 840ms

OS Name:Linux

OS Platform:x86_64

OS Release:3.10.0-229.4.2.el7.x86_64

OS Version:#1 SMP Fri Apr 24 15:26:38 EDT 2015

Compiler Version:unknown

3 tests failed.

Name	Status	Time	Details	Labels	Summary
Amesos2_Epetra_RowMatrix_Adapter_UnitTests_MPI_4	Failed	1s 860ms	Completed (Failed)	Amesos2	Broken
Amesos2_Epetra_MultiVector_Adapter_UnitTests_MPI_4	Failed	1s 980ms	Completed (Failed)	Amesos2	Broken
Amesos2_Tpetra_CrsMatrix_Adapter_UnitTests_MPI_4	Failed	1s 900ms	Completed (Failed)	Amesos2	Broken

Trilinos automated testing

- If you may have broken something, you will get an email about it

 CDash <trilinos-regression@sandia.gov> 4:05 AM (5 hours ago)   
to anasazi-regres. 

A submission to CDash for the project Trilinos has failing tests.
You have been identified as one of the authors who have checked in changes that are part of this submission or you are listed in the default contact list.

Details on the submission can be found at <http://testing.sandia.gov/cdash/buildSummary.php?buildid=2469557>

Project: Trilinos

SubProject: Anasazi

Site: artemis.srn.sandia.gov

Build Name: Linux-intel-15.0.2-MPI_RELEASE_DEV_DownStream_ETI_SERIAL-OFF_OPENMP-ON_PTHREAD-OFF_CUDA-OFF_COMPLEX-OFF

Build Time: 2016-06-06T03:59:42 MDT

Type: Nightly

Tests failing: 1

Tests failing

Anasazi_Epetra_MVOPTester_MPI_4 (<http://testing.sandia.gov/cdash/testDetails.php?test=33891492&build=2469557>)

How do you motivate somebody to write all those tests?



- Tests protect YOU from other people breaking your work
 - If someone else's changes break your code, they are responsible for fixing it
- You may already have some
 - Drivers for generating conference or paper results
 - Just reduce the problem size
 - User submitted bugs
 - Ask for a file that reproduces the issue
 - These make great regression tests
 - Examples
 - Add a pass/fail condition and you have a test

How do I determine what other tests I need?



- Code coverage tools
 - Expose parts of the code that aren't being tested
 - gcov
 - standard utility with the GNU compiler collection suite
 - counts the number of times each statement is executed
 - lcov
 - a graphical front-end for gcov
 - available at <http://ltp.sourceforge.net/coverage/lcov.php>

How to use lcov

- Compile and link your code with --coverage flag
 - It's a good idea to disable optimization
- Run your test suite
- Collect coverage data using lcov
- Generate html output using genhtml

A simple example

```
#include<iostream>
#include "isEven.hpp"

int main()
{
    int num = 8;

    if(isEven(num))
        std::cout << num << " is an even number.\nTEST PASSED";
    else
        std::cout << num << " is an odd number.\nTEST FAILED";

    return 0;
}

bool isEven(int x)
{
    if(x%2 == 0)
        return true;
    return false;
}
```

A simple example

- Compile and link with --coverage flag
 - `g++ --coverage evenExample.cpp -o evenExample`
 - This creates a file called `evenExample.gcno`
- Run the test
 - `./evenExample`
 - This creates a file called `evenExample.gcda`
- Collect coverage data using lcov
 - `lcov --capture --directory . --output-file evenExample.info`
 - This creates `evenExample.info`
- Generate html output using genhtml
 - `genhtml evenExample.info --output-directory evenHTML`
 - This generates html files in the directory `evenHTML`

A simple example

LCOV - code coverage report

Current view: [top level](#) - /home/amklinv/IDEAS/testingTalk/examples/simpleExample

Test: evenExample.info

Date: 2016-05-24 14:13:07

	Hit	Total	Coverage
Lines:	9	11	81.8 %
Functions:	4	4	100.0 %

Filename	Line Coverage		Functions	
evenExample.cpp		85.7 %	6 / 7	100.0 %
isEven.hpp		75.0 %	3 / 4	100.0 %

Generated by: [LCOV version 1.12-4-g04a3c0e](#)

This is the file we're testing



A simple example

LCOV - code coverage report

Current view: [top level](#) - home/amklinv/IDEAS/testingTalk/examples/simpleExample - isEven.hpp (source / [functions](#))

Test: evenExample.info

Date: 2016-05-24 14:13:07

	Hit	Total	Coverage
Lines:	3	4	75.0 %
Functions:	1	1	100.0 %

Line data Source code

1	1 : bool isEven(int x)
2	: {
3	1 : if(x%2 == 0)
4	1 : return true;
5	:
6	0 : return false; 
7	: }

We never tested this line of code
(which activates when x is odd)

Let's add another test

```
#include<iostream>
#include "isEven.hpp"

int main()
{
    int num = 7;

    if(isEven(num))
        std::cout << num << " is an even number.\nTEST FAILED";
    else
        std::cout << num << " is an odd number.\nTEST PASSED";

    return 0;
}

bool isEven(int x)
{
    if(x%2 == 0)
        return true;
    return false;
}
```

A simple example

- Compile and link with --coverage flag
 - `g++ --coverage oddExample.cpp -o oddExample`
 - This creates a file called `oddExample.gcno`
- Run the test
 - `./oddExample`
 - This creates a file called `oddExample.gcda`
- Collect coverage data for BOTH TESTS using lcov
 - `lcov --capture --directory . --output-file twoExamples.info`
 - This creates `twoExamples.info`
- Generate html output using genhtml
 - `genhtml twoExamples.info --output-directory totalHTML`
 - This generates html files in the directory `totalHTML`

A simple example

LCOV - code coverage report

Current view:	top level	- /home/amklinv/IDEAS/testingTalk/examples/simpleExample	Hit	Total	Coverage
Test:	twoExamples.info		Lines:	16	18 88.9 %
Date:	2016-05-24 15:17:38		Functions:	7	7 100.0 %

Filename	Line Coverage	Functions
evenExample.cpp	 85.7 %	6 / 7 100.0 % 3 / 3
isEven.hpp	 100.0 %	4 / 4 100.0 % 1 / 1
oddExample.cpp	 85.7 %	6 / 7 100.0 % 3 / 3

Generated by: [LCOV version 1.12-4-g04a3c0e](#)

This is the file we're testing

A simple example

LCOV - code coverage report

Current view: [top level](#) - [home/amklinv/IDEAS/testingTalk/examples/simpleExample](#) - `isEven.hpp` (source / functions)

Test: `twoExamples.info`

Date: `2016-05-24 15:17:38`

	Hit	Total	Coverage
Lines:	4	4	100.0 %
Functions:	1	1	100.0 %

Line data Source code

1	2 : bool isEven(int x)
2	: {
3	2 : if(x%2 == 0)
4	1 : return true;
5	:
6	1 : return false;
7	: }

We tested every line of this function

A real example - xSDKTrilinos

- Part of the Trilinos library, developed at SNL as part of the IDEAS project
- Contains the interfaces between Trilinos, PETSc, and hypre
- Available at <https://github.com/trilinos/xSDKTrilinos>
- Ten automated tests are run nightly
 - Six are actually examples that were converted into tests
- Did we leave anything out?

A real example - xSDKTrilinos

- Step 1: Modify our CMake configuration file to use the --coverage flag to compile and link

```
trilinos-build : vim
File Edit View Scrollback Bookmarks Settings Help
\\
-D TPL_ENABLE_PETSC:BOOL=ON \
-D PETSC_LIBRARY_DIRS:FILEPATH="${PETSC_LIB_DIR}" \
-D PETSC_INCLUDE_DIRS:FILEPATH="${PETSC_INCLUDE_DIR}" \
\\
-D TPL_ENABLE_ParMETIS:BOOL=ON \
-D ParMETIS_LIBRARY_DIRS:FILEPATH="${SUPERLU_LIB_DIR}" \
-D ParMETIS_INCLUDE_DIRS:FILEPATH="${SUPERLU_INCLUDE_DIR}" \
\\
-D TPL_ENABLE_HYPRE:BOOL=ON \
-D HYPRE_LIBRARY_DIRS:FILEPATH="${HYPRE_LIB_DIR}" \
-D HYPRE_INCLUDE_DIRS:FILEPATH="${HYPRE_INCLUDE_DIR}" \
\\
-D TPL_ENABLE_SuperLUDist:BOOL=ON \
-D SuperLUDist_LIBRARY_DIRS:FILEPATH="${SUPERLU_LIB_DIR}" \
-D SuperLUDist_INCLUDE_DIRS:FILEPATH="${SUPERLU_INCLUDE_DIR}" \
\\
-D Trilinos_ENABLE_Amesos2:BOOL=ON \
-D Trilinos_ENABLE_xSDKTrilinos:BOOL=ON \
\\
-D CMAKE_CXX_FLAGS:STRING="--coverage" \
-D CMAKE_C_FLAGS:STRING="--coverage" \
-D CMAKE_EXE_LINKER_FLAGS:STRING="--coverage" \
-D Trilinos_ENABLE_Fortran:BOOL=OFF \
\\
${TRILINOS_HOME}
-- INSERT --
```

A real example - xSDKTrilinos

- Build Trilinos (including xSDKTrilinos)
 - ./do-configure
 - make -j
- This will create a whole bunch of .gcno files
- This will also build the xSDKTrilinos tests because the configure file included
 - -D Trilinos_ENABLE_TESTS:BOOL=ON
 - -D Trilinos_ENABLE_EXAMPLES:BOOL=ON
 - -D Trilinos_ENABLE_ALL_OPTIONAL_PACKAGES=ON

A real example - xSDKTrilinos

- Run the tests using `ctest`
- Note that this is not prohibitively slow

trilinos-build : ctest

```
[amklinv@s995692 trilinos-build]$ ctest
Test project /home/amklinv/IDEAS/testingTalk/trilinos-build
  Start 1: Amesos2_KLU2_UnitTests_MPI_4
1/18 Test #1: Amesos2_KLU2_UnitTests_MPI_4 ..... Passed    1.46 sec
  Start 2: Amesos2_SuperLU_DIST_Solver_Test_MPI_4
2/18 Test #2: Amesos2_SuperLU_DIST_Solver_Test_MPI_4 ..... Passed    2.80 sec
  Start 3: Amesos2_SolverFactory_UnitTests_MPI_4
3/18 Test #3: Amesos2_SolverFactory_UnitTests_MPI_4 ..... Passed    1.46 sec
  Start 4: Amesos2_Tpetra_MultiVector_Adapter_UnitTests_MPI_4
4/18 Test #4: Amesos2_Tpetra_MultiVector_Adapter_UnitTests_MPI_4 ... Passed    1.36 sec
  Start 5: Amesos2_Tpetra_CrsMatrix_Adapter_UnitTests_MPI_4
5/18 Test #5: Amesos2_Tpetra_CrsMatrix_Adapter_UnitTests_MPI_4 .... Passed    1.42 sec
  Start 6: Amesos2_Epetra_MultiVector_Adapter_UnitTests_MPI_4
6/18 Test #6: Amesos2_Epetra_MultiVector_Adapter_UnitTests_MPI_4 ... Passed    1.35 sec
  Start 7: Amesos2_Epetra_RowMatrix_Adapter_UnitTests_MPI_4
7/18 Test #7: Amesos2_Epetra_RowMatrix_Adapter_UnitTests_MPI_4 .... Passed    1.35 sec
  Start 8: Amesos2_CrsMatrix_Adapter_Conistency_Tests_MPI_4
8/18 Test #8: Amesos2_CrsMatrix_Adapter_Conistency_Tests_MPI_4 .... Passed    1.47 sec
  Start 9: xSDKTrilinos_PETScAIJMatrix_MPI_4
9/18 Test #9: xSDKTrilinos_PETScAIJMatrix_MPI_4 ..... Passed    1.42 sec
  Start 10: xSDKTrilinos_PETSc_Amesos2_example_MPI_4
10/18 Test #10: xSDKTrilinos_PETSc_Amesos2_example_MPI_4 ..... Passed    1.42 sec
  Start 11: xSDKTrilinos_PETSc_Anasazi_example_MPI_4
11/18 Test #11: xSDKTrilinos_PETSc_Anasazi_example_MPI_4 ..... Passed    2.71 sec
  Start 12: xSDKTrilinos_PETSc_Itpack2_example_MPI_4
12/18 Test #12: xSDKTrilinos_PETSc_Itpack2_example_MPI_4 ..... Passed    1.47 sec
  Start 13: xSDKTrilinos_PETSc_MueLu_example_MPI_4
```

...ilinos-build : ctest

A real example - xSDKTrilinos

- All tests passed. Yay!
 - This also created a bunch of .gcda files

```

trilinos-build : ctest
File Edit View Scrollback Bookmarks Settings Help
Start 10: xSDKTrilinos_PETSc_Amesos2_example_MPI_4
10/18 Test #10: xSDKTrilinos_PETSc_Amesos2_example_MPI_4 ..... Passed 1.42 sec
Start 11: xSDKTrilinos_PETSc_Anasazi_example_MPI_4
11/18 Test #11: xSDKTrilinos_PETSc_Anasazi_example_MPI_4 ..... Passed 2.71 sec
Start 12: xSDKTrilinos_PETSc_Itpack2_example_MPI_4
12/18 Test #12: xSDKTrilinos_PETSc_Itpack2_example_MPI_4 ..... Passed 1.47 sec
Start 13: xSDKTrilinos_PETSc_MueLu_example_MPI_4
13/18 Test #13: xSDKTrilinos_PETSc_MueLu_example_MPI_4 ..... Passed 2.34 sec
Start 14: xSDKTrilinos_example_TpetraKSP_MPI_4
14/18 Test #14: xSDKTrilinos_example_TpetraKSP_MPI_4 ..... Passed 1.50 sec
Start 15: xSDKTrilinos_example_EpetraKSP_MPI_4
15/18 Test #15: xSDKTrilinos_example_EpetraKSP_MPI_4 ..... Passed 1.37 sec
Start 16: xSDKTrilinos_HyPreTest_MPI_4
16/18 Test #16: xSDKTrilinos_HyPreTest_MPI_4 ..... Passed 1.42 sec
Start 17: xSDKTrilinos_HyPre_Belos_example_MPI_4
17/18 Test #17: xSDKTrilinos_HyPre_Belos_example_MPI_4 ..... Passed 1.38 sec
Start 18: xSDKTrilinos_HyPre_Solve_example_MPI_4
18/18 Test #18: xSDKTrilinos_HyPre_Solve_example_MPI_4 ..... Passed 1.36 sec

100% tests passed, 0 tests failed out of 18

Label Time Summary:
Amesos2      = 12.67 sec (8 tests)
xSDKTrilinos = 16.39 sec (10 tests)

Total Test time (real) = 29.11 sec
[amklinv@s995692 trilinos-build]$ █

```

A real example - xSDKTrilinos

- Collect coverage data for the tests using lcov
 - `lcov --capture --directory . --output-file xSDKTrilinos.info`
 - This creates `xSDKTrilinos.info`
 - lcov processes 634 gcda files in this step, so this does take a few minutes

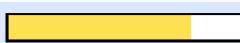
A real example - xSDKTrilinos

- Generate html output using genhtml
 - `genhtml xSDKTrilinos.info --output-directory xSDKTrilinos`
 - This generates html files in the directory xSDKTrilinos
 - This step takes a few minutes too

A real example - xSDKTrilinos

LCOV - code coverage report

Current view:	top level - xSDKTrilinos/petsc/src	Hit	Total	Coverage
Test:	xSDKTrilinos.info	Lines: 342	420	81.4 %
Date:	2016-06-02 15:36:10	Functions: 77	117	65.8 %

Filename	Line Coverage	Functions
BelosPETScSolMgr.hpp	 84.7 %	166 / 196 68.2 %
Tpetra_PETScAIJGraph.hpp	 75.3 %	67 / 89 62.5 %
Tpetra_PETScAIJMatrix.hpp	 80.7 %	109 / 135 65.9 %

Generated by: [LCOV version 1.12-4-g04a3c0e](#)

Let's take a look at the solver interface.

```
766 : //=====
767 : template<class ScalarType, class MV, class OP>
768 : PetscErrorCode PETScSolMgr<ScalarType,MV,OP>::applyPrec(PC M, Vec x, Vec Mx)
769 : {
770 :     using Teuchos::RCP;
771 :     typedef PETScSolMgrHelper<ScalarType,MV,OP> Helper;
772 :
773 :     PetscErrorCode ierr;
774 :     const PetscScalar * xData;
775 :     PetscScalar * MxData;
776 :     void * ptr;
777 :
778 :     // Get the problem out of the context
779 :     ierr = PCShellGetContext(M,&ptr); CHKERRQ(ierr);
780 :     LinearProblem<ScalarType,MV,OP> * problem = (LinearProblem<ScalarType,MV,OP>*)ptr;
781 :
782 :     // Rip the raw data out of the PETSc vectors
783 :     ierr = VecGetArrayRead(x, &xData); CHKERRQ(ierr);
784 :     ierr = VecGetArray(Mx, &MxData); CHKERRQ(ierr);
785 :
786 :     // Wrap the PETSc data in a Trilinos Vector
787 :     RCP<MV> trilinosX, trilinosMX;
788 :     Helper::wrapVector(const_cast<PetscScalar*>(xData), *problem->getLHS(), trilinosX);
789 :     Helper::wrapVector(MxData, *problem->getLHS(), trilinosMX);
790 :
791 :     // Perform the multiplication
792 :     if(problem->isLeftPrec()) {
793 :         problem->applyLeftPrec(*trilinosX, *trilinosMX);
794 :     }
795 :     else {
796 :         problem->applyRightPrec(*trilinosX, *trilinosMX);
797 :     }
798 :
799 :     // Unwrap the vectors; this is necessary if we copied data in the wrap step
800 :     Helper::unwrapVector(MxData, trilinosMX);
801 :
802 :     // Restore the PETSc vectors
803 :     ierr = VecRestoreArrayRead(x,&xData); CHKERRQ(ierr);
804 :     ierr = VecRestoreArray(Mx,&MxData); CHKERRQ(ierr);
805 :
806 :     192 :     return 0;
807 :     : }
```

A real example - xSDKTrilinos

```
791      :    // Perform the multiplication
792      192 :    if(problem->isLeftPrec()) {
793      192 :      problem->applyLeftPrec(*trilinosX, *trilinosMX);
794      :    }
795      :    else {
796      0 :      problem->applyRightPrec(*trilinosX, *trilinosMX);
797      :    }
```



Oops. I never tested the RIGHT preconditioning branch.

Code coverage disclaimer

100% code coverage does not ensure bug-free code

VERIFICATION

Why is verification a separate topic?



- Code verification *uses* tests
- It is much more than a collection of tests
- It is the holistic process through which you ensure that
 - your implementation shows expected behavior,
 - your implementation is consistent with your model,
 - science you are trying to do with the code can be done.

Many stages and types of verification



- During initial code development
 - accuracy and stability during development of the algorithm
 - matching the algorithm to the model
 - interoperability of algorithms
- In later stages
 - Ongoing maintenance
 - while adding new major capabilities or modifying existing capabilities
 - Preparing for production
- If refactoring
 - Ensuring that behavior remains consistent and expected
- All stages have a mix of automation and human-intervention

Development Phase

- Development of tests and diagnostics goes hand-in-hand with code development
 - Non-trivial to devise good tests, but extremely important
 - A code is only as good as its tests
 - Compare against simpler analytical or semi-analytical solutions
 - They can also form a basis for unit testing
- In addition to testing for “correct” behavior, also test for stability, convergence, or other such desirable characteristics
- Many of these tests will be worth preserving for the maintenance phase

Mature Phase

- A subset of tests developed during the development phase become part of the regular testing regime
 - Focus on both code and functionality coverage
 - Code coverage by itself does not guarantee correctness if the many moving parts also do not interoperate well
 - Tweak in one part may result in correct behavior of individual parts, but not when they work together
- When new features or capabilities are added to the code -
 - Some tests will have been generated during development
 - The capability should have been verified for interoperability with existing code
 - A subset of all those tests should get included in the test-suite
 - Following the same principles of code and functionality coverage

Other Phases

- Preparing for simulations
 - Targeted testing of production configuration
 - Performance testing and tuning
 - More on this tomorrow...
- Refactoring
 - Possible expansion of test-suite
 - Numerical drift in results
 - Ramp-on planning
 - More on this tomorrow...

Selecting Tests

- Selection of tests is non-trivial
 - Tools exist for code coverage
 - Not for interoperability coverage
- One approach : use a matrix
 - Put infrastructure components in rows, science components in columns
 - List interoperability constraints, and pick tests
 - Tests for ongoing productions
 - Tests known to be sensitive to perturbations
 - Least complex tests that can cover the empty spots
 - Least complex tests that meet the missing interoperability constraints

Dubey et al, *Ongoing verification of a multiphysics community code: FLASH*,
Software: Practice and Experience Vol 45(2) pp. 233-244

Example from FLASH

	Hydro	EOS	Gravity	Burn	Particles
AMR	CL	CL		CL	CL
UG	SV	SV			SV
Multigrid	WD	WD	WD	WD	
FFT			PT		

Tests	Symbol
Sedov	SV
Cellular	CL
Poisson	PT
White Dwarf	WD

- A test on the same row indicates interoperability between corresponding physics
- Similar logic would apply to tests on the same column for infrastructure
- More goes on, but this is the primary methodology

You can pick rows and columns in many different ways

Hands on session tonight

- A small linear algebra package called Morpheus
 - Contains matrix and vector classes and functions
- We will...
 - Examine its Doxygen documentation
 - Determine the code coverage using gcov
 - Discuss how to improve its test suite