Kitware

Building Science with CMake

Bill Hoffman bill.hoffman@kitware.com

Bill Hoffman

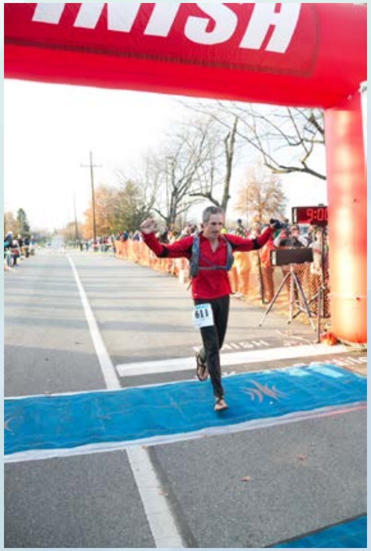
- One of 5 founders of Kitware Inc
- Originator of CMake build tool
- Ultra runner (barefoot/sandals)

Kitware

CMake/CPack/CTest/CDash

Open Source Tools to Build Test and Deploy C++ Software

Bill Hoffman
bill.hoffman@kitware.com



Kitware

Collaborative Software R&D

Algorithms & applications

Software process & infrastructure

Support & training

Open source leadership

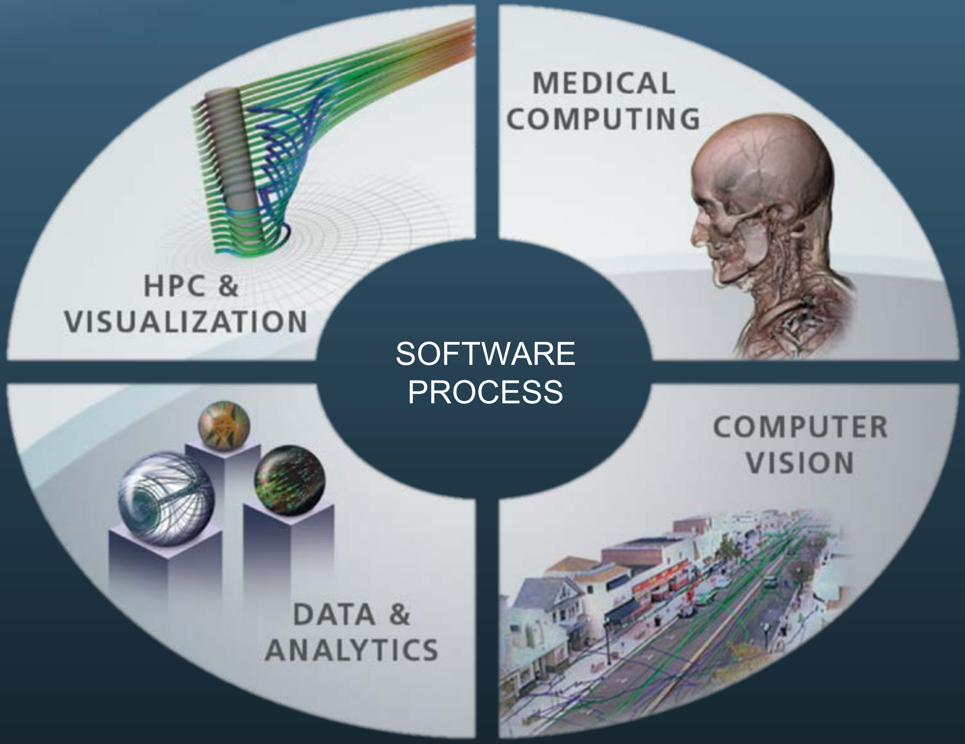
Supporting all sectors

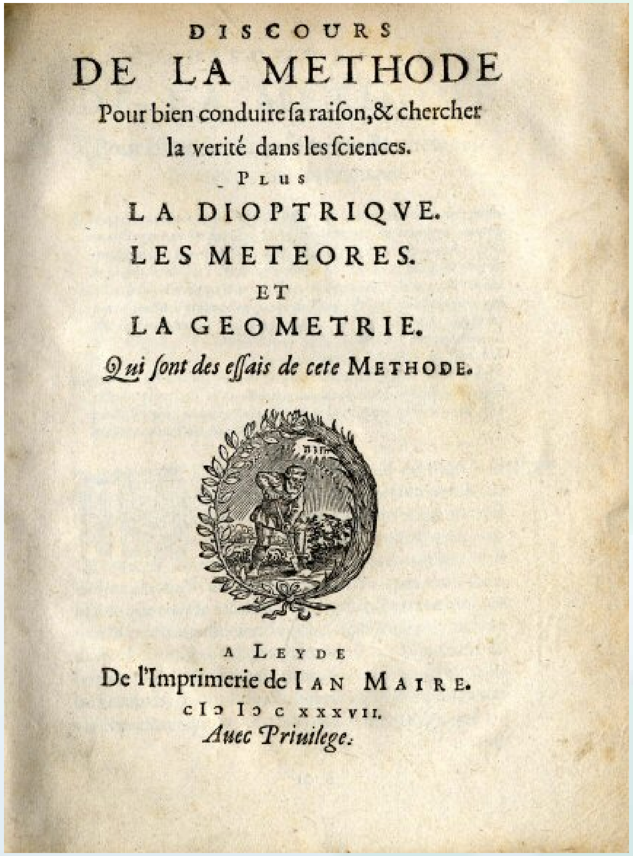
Industry, government & academia

Successful small business

Founded in 1998; privately owned

Offices in NY, NC, NM & Lyon France





DISCOURS
DE LA METHODE
Pour bien conduire ſa raiſon, & chercher
la verité dans les ſciences.
PLUS
LA DIOPTRIQVE.
LES METEORES.
ET
LA GEOMETRIE.
Qui ſont des eſſais de cete METHODE.

A LEYDE
De l'Imprimerie de IAN MAIRE.
CIƆ IƆ C XXXVII.
Auec Priuilege.

Discourse on the (Scientific) Method*, Descartes 1637*

DOUBTING EVERYTHING, and only believe in those things that are evidently true (REPRODUCIBLE)

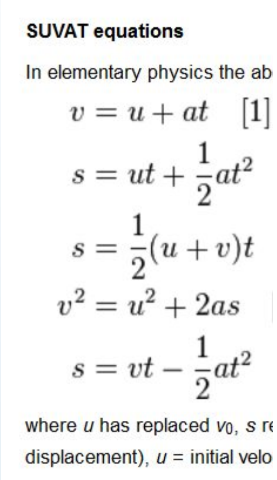
If it's not reproducible, it's not Science

Nullius in Verba

"take nobody's word for it"
Royal Society 1640

Scien   ation

- “Softw
  mathe
  langua
  Seidel



SUVAT equations

In elementary physics the ab

$$v = u + at \quad [1]$$

$$s = ut + \frac{1}{2}at^2$$

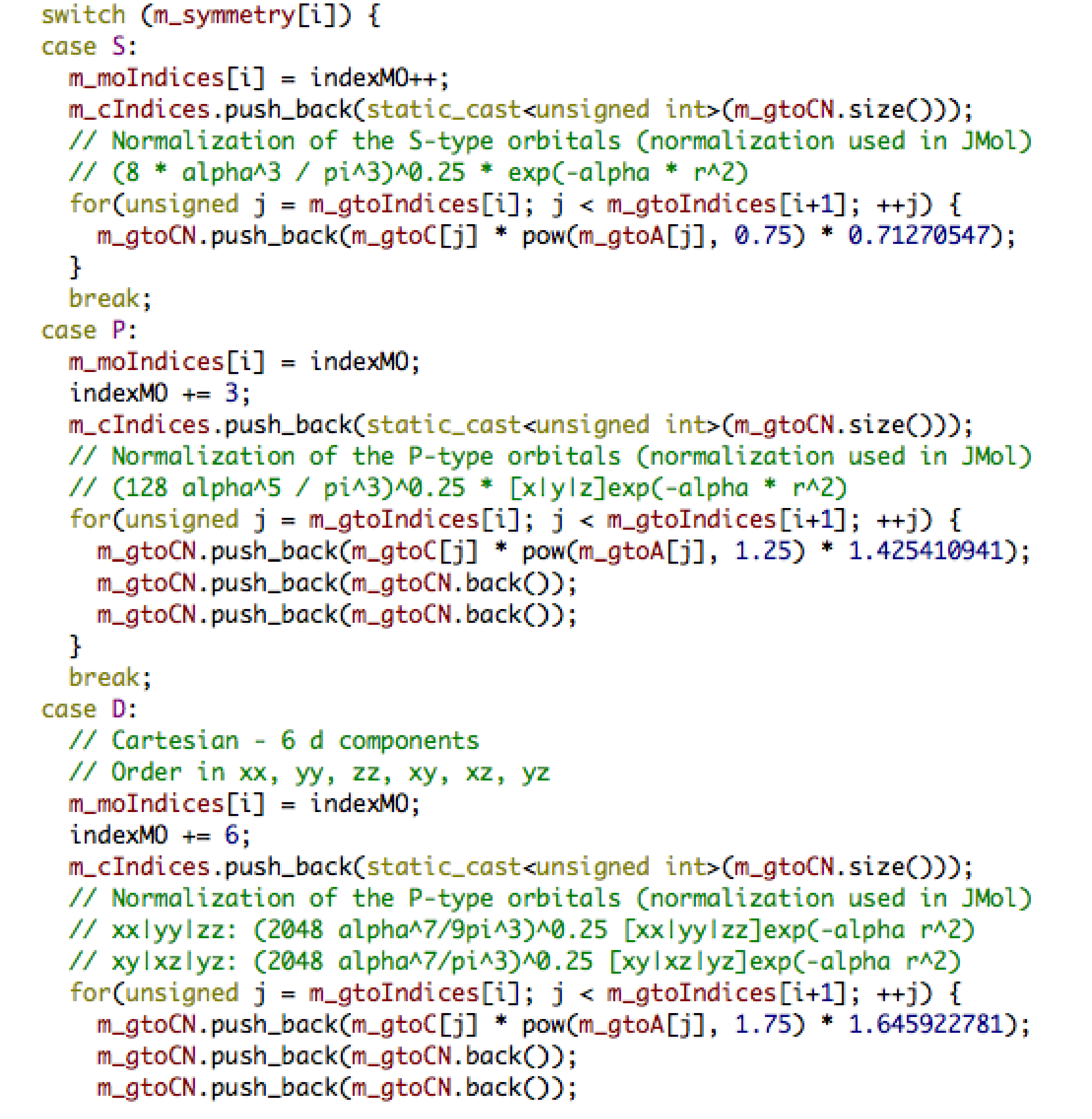
$$s = \frac{1}{2}(u+v)t$$

$$v^2 = u^2 + 2as$$

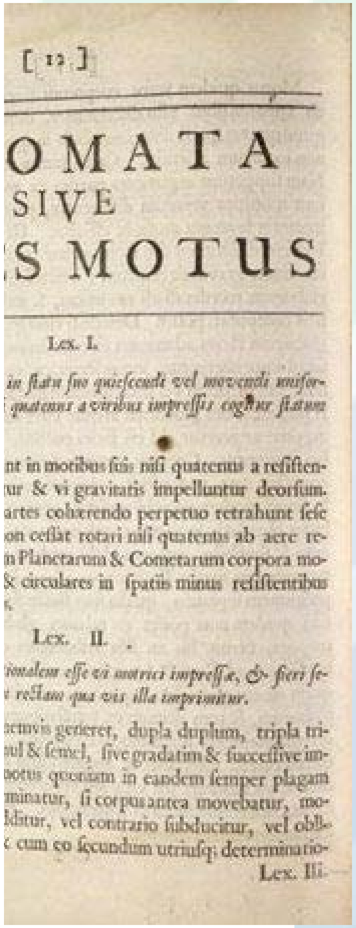
$$s = vt - \frac{1}{2}at^2$$

where u has replaced v_0, s re

displacement), u = initial velo



```
switch (m_symmetry[i]) {
case S:
  m_moIndices[i] = indexMO++;
  m_cIndices.push_back(static_cast<unsigned int>(m_gtoCN.size()));
  // Normalization of the S-type orbitals (normalization used in JMol)
  // (8 * alpha^3 / pi^3)^0.25 * exp(-alpha * r^2)
  for(unsigned j = m_gtoIndices[i]; j < m_gtoIndices[i+1]; ++j) {
    m_gtoCN.push_back(m_gtoC[j] * pow(m_gtoA[j], 0.75) * 0.71270547);
  }
  break;
case P:
  m_moIndices[i] = indexMO;
  indexMO += 3;
  m_cIndices.push_back(static_cast<unsigned int>(m_gtoCN.size()));
  // Normalization of the P-type orbitals (normalization used in JMol)
  // (128 alpha^5 / pi^3)^0.25 * [x|y|z]exp(-alpha * r^2)
  for(unsigned j = m_gtoIndices[i]; j < m_gtoIndices[i+1]; ++j) {
    m_gtoCN.push_back(m_gtoC[j] * pow(m_gtoA[j], 1.25) * 1.425410941);
    m_gtoCN.push_back(m_gtoCN.back());
    m_gtoCN.push_back(m_gtoCN.back());
  }
  break;
case D:
  // Cartesian - 6 d components
  // Order in xx, yy, zz, xy, xz, yz
  m_moIndices[i] = indexMO;
  indexMO += 6;
  m_cIndices.push_back(static_cast<unsigned int>(m_gtoCN.size()));
  // Normalization of the P-type orbitals (normalization used in JMol)
  // xx|yy|zz: (2048 alpha^7/9pi^3)^0.25 [xx|yy|zz]exp(-alpha r^2)
  // xy|xz|yz: (2048 alpha^7/pi^3)^0.25 [xy|xz|yz]exp(-alpha r^2)
  for(unsigned j = m_gtoIndices[i]; j < m_gtoIndices[i+1]; ++j) {
    m_gtoCN.push_back(m_gtoC[j] * pow(m_gtoA[j], 1.75) * 1.645922781);
    m_gtoCN.push_back(m_gtoCN.back());
    m_gtoCN.push_back(m_gtoCN.back());
```



[12]

OMATA
SIVE
S MOTUS

Lex. I.

in ſtatu ſuo quieſcendi vel movendi unifor-
quatenus a viribus impreſſis cogitur ſtatum

nt in motibus ſuis niſi quatenus a reſiſten-
ur & vi gravitatis impelluntur deorſum.
artes cohærendo perpetuo retrahunt ſeſe
on ceſſat rotari niſi quatenus ab aere re-
n Planetarum & Cometarum corpora mo-
& circulares in ſpatiis minus reſiſtentibus

Lex. II.

ionalem eſſe vi motrici impreſſæ, & fieri ſe-
rectam qua vis illa imprimitur.

emvis generet, dupla duplum, tripla tri-
ul & ſemel, ſive gradatim & ſucceſſive im-
otus quoniam in eandem ſemper plagam
minatur, ſi corpus antea movebatur, mo-
dditur, vel contrario ſubducitur, vel obli-
cum eo ſecundum utriuſq; determinatio-
Lex. III.

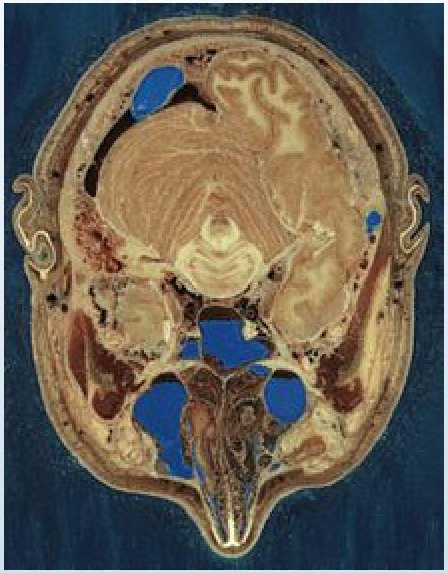
Failure of Reproducibility

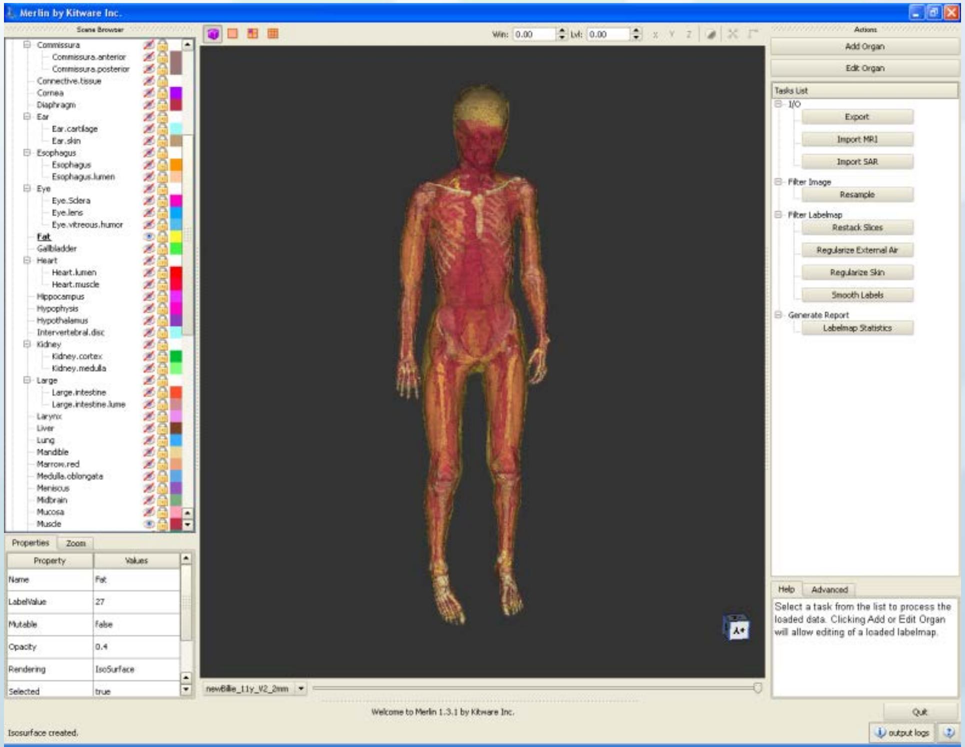
- ***Nature*** *(March 2012)*
  - Glenn Begley, former head of cancer research at pharma giant Amgen
  - Lee M. Ellis, cancer researcher at the University of Texas

Found that more than <u>90% of papers</u> published in science journals describing "landmark" breakthroughs in preclinical cancer research, <u>are not reproducible</u>, and are thus just plain wrong.

CMake came from open science

- NIH Visible Human Project
  - Data, CT/MR/Slice 1994/1995
  - Code (ITK) 1999
    - CMake – Cross platform build tool





Win: 0.00 Lvl: 0.00

Property	Values
Name	Fat
LabelValue	27
Mutable	false
Opacity	0.4
Rendering	IsoSurface
Selected	true

newBillie_11y_V2_2mm

Welcome to Merlin 1.3.1 by Kitware Inc.

Isosurface created.

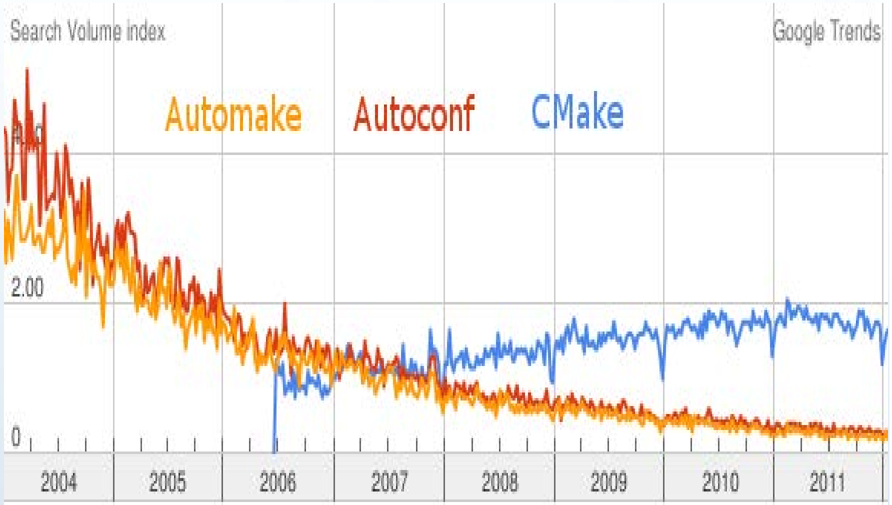
Select a task from the list to process the loaded data. Clicking Add or Edit Organ will allow editing of a loaded labelmap.

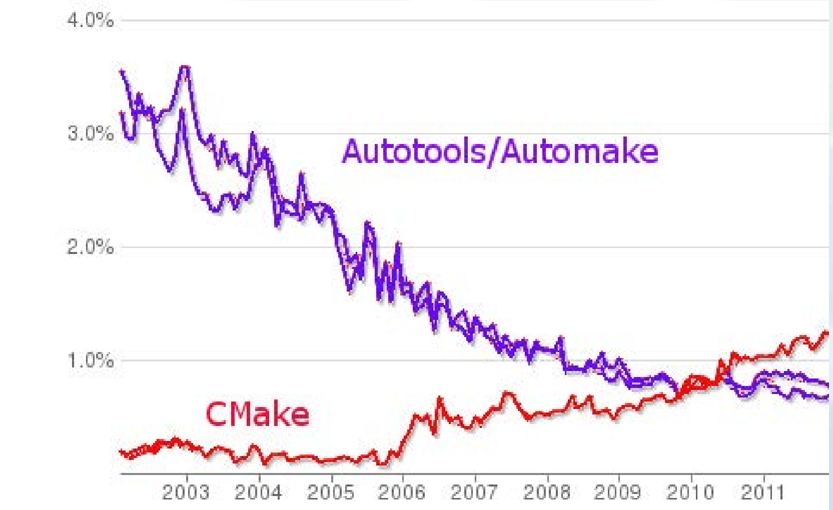
How CMake Changes The Way We Build C++

- Boost aims to give C++ a set of useful libraries like Java, Python, and C#
- CMake aims to give C++ compile portability like the compile once and run everywhere of Java, Python, and C#
  - Same build tool and files for all platforms
  - Easy to mix both large and small libraries

CMake most popular Kitware Tool

KDE 2006 – Tipping Point!

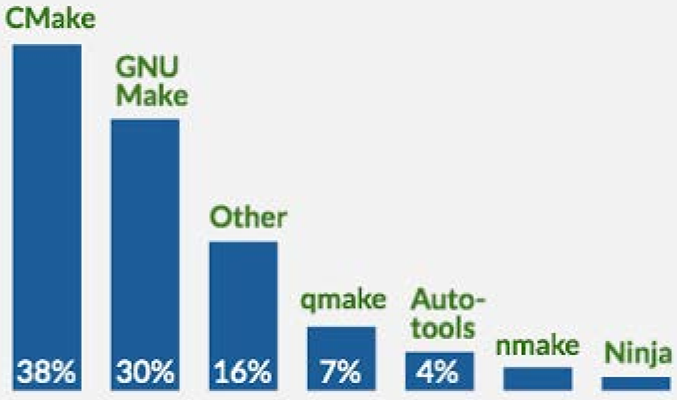




- Google Search Trends and ohloh comparisons with auto*
- 5000+ downloads per day from www.cmake.org
- Major Linux distributions and Cygwin provide CMake packages
- KDE, Second Life, CLang, many others

CMake is the most popular build tool at 38%.

- 2014 study data by JetBrains in the development of a new C++ IDE, that is CLion found CMake to have the highest adoption by the 4.4 million worldwide C++ developer community



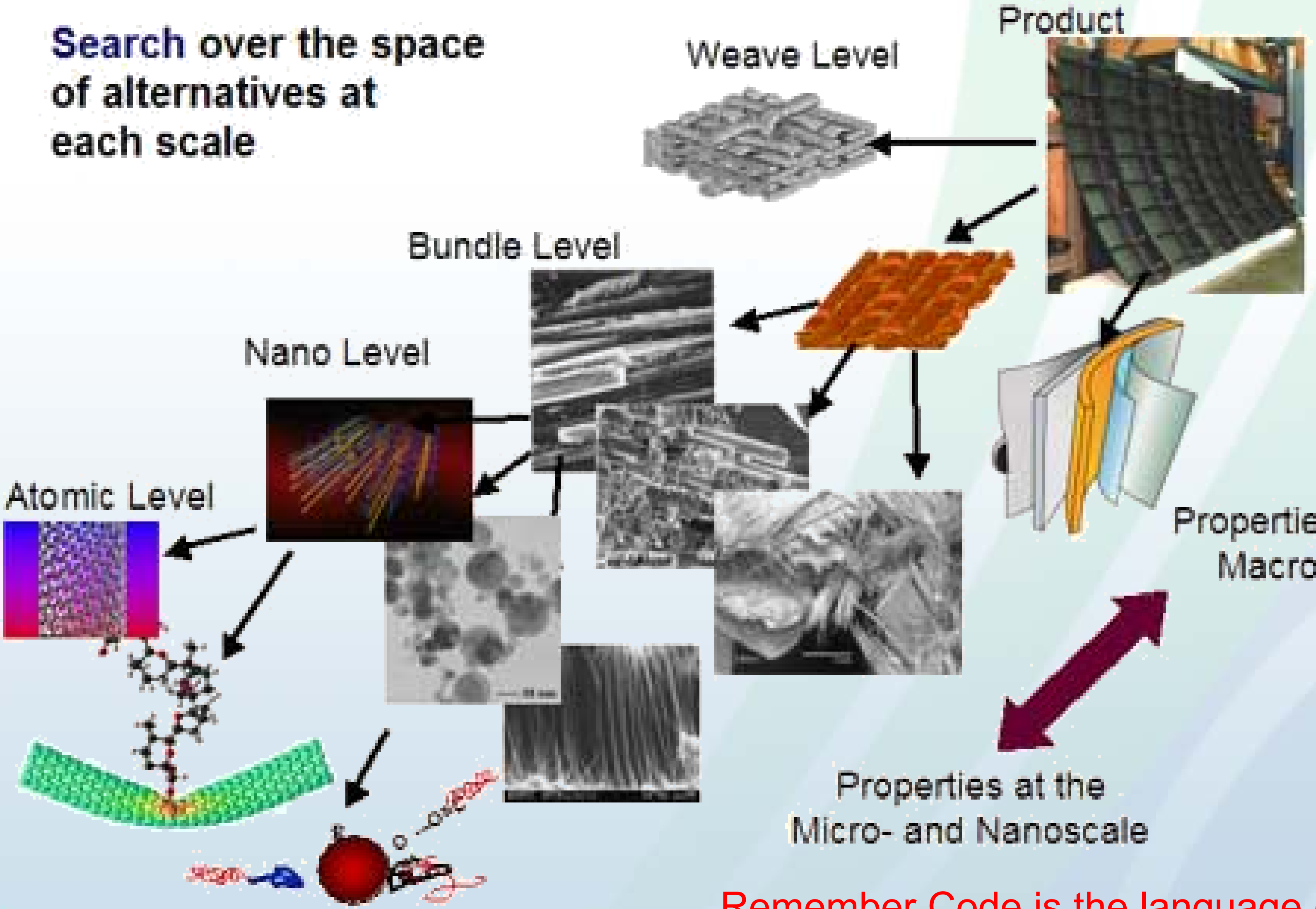
Popular C++ build systems

CMake and GNU Make build systems are the two close leaders, together accounting for 68% of all C++ developers.

Science is not done by one person and problems bigger

Multiscale Design

Search over the space of alternatives at each scale



Product

Weave Level

Bundle Level

Nano Level

Atomic Level

Properties at the Macroscale

Properties at the Micro- and Nanoscale

Remember Code is the language of Science

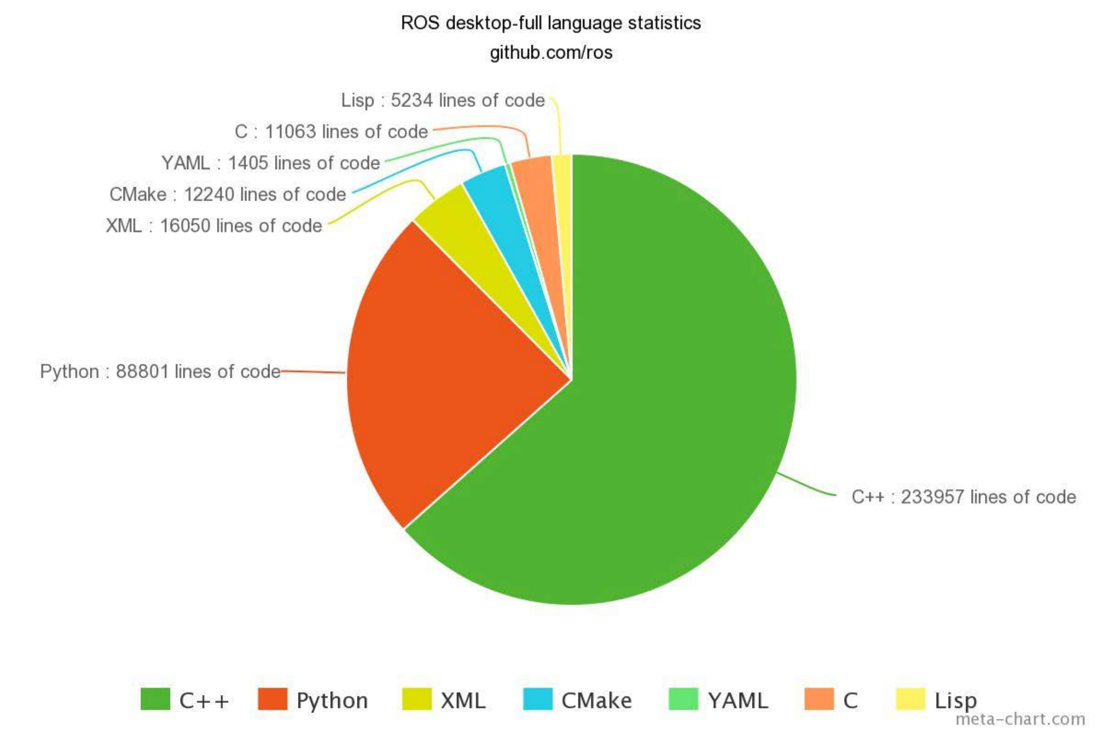
Courtesy SCOREC RPI

Robot Operating System (ROS) Statistics

Metrics as of July 2015 (http://wiki.ros.org/Metrics):

- Unique IPs downloading ROS debs: ~45,000/month
- Academic papers citing original paper: 1843
- Robot models officially supported: >101
- wiki.ros.org pageviews: ~37,000/day

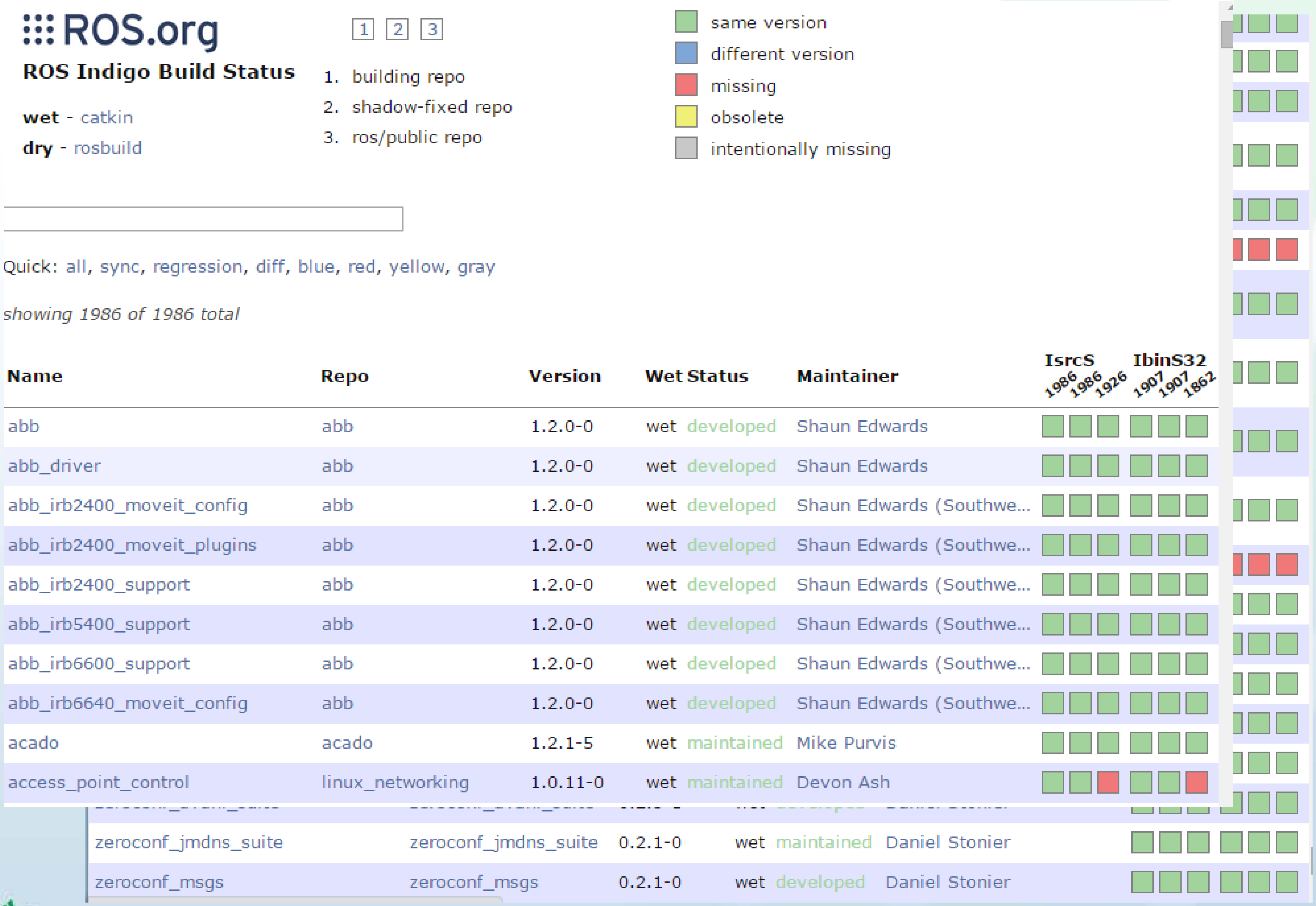
Longest distance a ROS robot has traveled from Earth: 270 miles



ROS is BIG

- Almost 2000 packages in our Indigo release: http://www.ros.org/debbuild/indigo.html
- All of those packages, as far as I know, are CMake projects or at least wrapped in CMake.
- They are contributed by roughly 370 different people/groups, based on the contributors to our releases repo: https://github.com/ros/rosdistro
- CMake makes up ~6% (>12kLoC) of the ROS Indigo code base (for desktop-full which is a subset and what we consider the core).

2000 CMake based packages in ROS



ROS.org

ROS Indigo Build Status

wet - catkin

dry - rosbuild

1 2 3

1. building repo
2. shadow-fixed repo
3. ros/public repo

🟩 same version

🟦 different version

🟥 missing

🟨 obsolete

⬜ intentionally missing

Quick: all, sync, regression, diff, blue, red, yellow, gray

showing 1986 of 1986 total

Name	Repo	Version	Wet Status	Maintainer	IsrcS			IbinS32		
					1986	1986	1926	1907	1907	1862
abb	abb	1.2.0-0	wet developed	Shaun Edwards	🟩	🟩	🟩	🟩	🟩	🟩
abb_driver	abb	1.2.0-0	wet developed	Shaun Edwards	🟩	🟩	🟩	🟩	🟩	🟩
abb_irb2400_moveit_config	abb	1.2.0-0	wet developed	Shaun Edwards (Southwe...	🟩	🟩	🟩	🟩	🟩	🟩
abb_irb2400_moveit_plugins	abb	1.2.0-0	wet developed	Shaun Edwards (Southwe...	🟩	🟩	🟩	🟩	🟩	🟩
abb_irb2400_support	abb	1.2.0-0	wet developed	Shaun Edwards (Southwe...	🟩	🟩	🟩	🟩	🟩	🟩
abb_irb5400_support	abb	1.2.0-0	wet developed	Shaun Edwards (Southwe...	🟩	🟩	🟩	🟩	🟩	🟩
abb_irb6600_support	abb	1.2.0-0	wet developed	Shaun Edwards (Southwe...	🟩	🟩	🟩	🟩	🟩	🟩
abb_irb6640_moveit_config	abb	1.2.0-0	wet developed	Shaun Edwards (Southwe...	🟩	🟩	🟩	🟩	🟩	🟩
acado	acado	1.2.1-5	wet maintained	Mike Purvis	🟩	🟩	🟩	🟩	🟩	🟩
access_point_control	linux_networking	1.0.11-0	wet maintained	Devon Ash	🟩	🟩	🟥	🟩	🟩	🟥
zeroconf_jmdns_suite	zeroconf_jmdns_suite	0.2.1-0	wet maintained	Daniel Stonier	🟩	🟩	🟩	🟩	🟩	🟩
zeroconf_msgs	zeroconf_msgs	0.2.1-0	wet developed	Daniel Stonier	🟩	🟩	🟩	🟩	🟩	🟩

Kitware

Why CMake? It's easy, and works well

- A build system that just works

- A build system that is easy to use cross platform

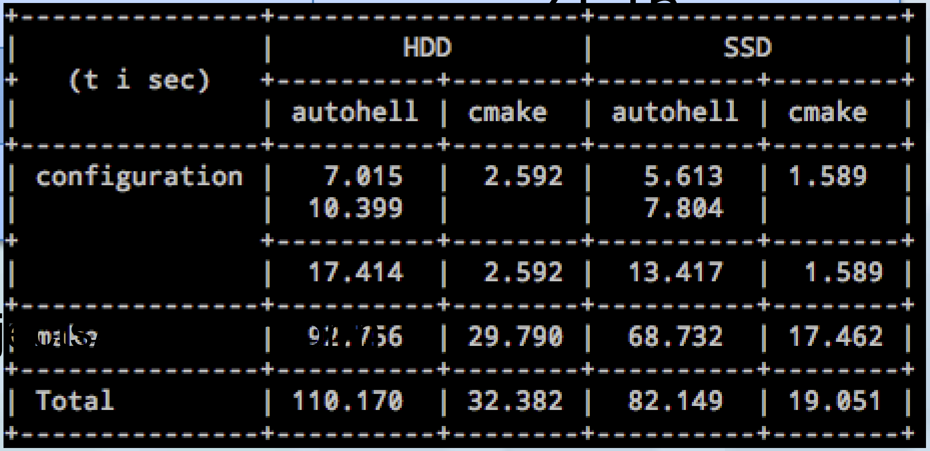
Typical Project without CMake (curl)

```
$ ls
CHANGES          RELEASE-NOTES  curl-config.in  missing
CMake            acinclude.m4   curl-style.el   mkinstalldirs
CMakeLists.txt     aclocal.m4     depcomp        notes
build       docs        notes~
COPYING            buildconf      include        packages
CVS              buildconf.bat  install-sh      reconf
ChangeLog          compile       lib            sample.emacs
Makefile           config.guess  libcurl.pc.in  src
Makefile.am          config.sub    ltmain.sh      tests
Makefile.in          configure     m4             vc6curl.dsw
README             configure.ac  maketgz

$ ls src/
CMakeLists.txt   Makefile.riscos  curlsrc.dsp  hugehelp.h     version.h
CVS          Makefile.vc6    curlsrc.dsw  macos         writeenv.c
Makefile.Watcom  Makefile.vc8     curlutil.c  main.c        writeenv.h
Makefile.am      config-amigaos.h  curlutil.h  makefile.amiga  writeout.c
Makefile.b32     config-mac.h     getpass.c   makefile.dj     writeout.h
Makefile.in      config-riscos.h  getpass.h   mkhelp.pl
Makefile.inc     config-win32.h   homedir.c   setup.h
Makefile.m32     config.h.in      homedir.h   urlglob.c
Makefile.netware  curl.rc         hugehelp.c  urlglob.h
```

Why CMake? It’s fast

http://blog.qgis.org/?q=node/16 : “I was quite surprised with the speed of building Quantum GIS codebase in comparison to Autotools. “

Task	CMake	Autotools
Configure	0:08	Automake: 0:41 Configure: 0:20
Make	12:15	21:16
Install	0:20	
Total	12:43	



```
+---------------+-------------------+-------------------+
|               |        HDD        |        SSD        |
+  (t i sec)    +----------+--------+----------+--------+
|               | autohell | cmake  | autohell | cmake  |
+---------------+----------+--------+----------+--------+
| configuration |    7.015 |  2.592 |    5.613 | 1.589  |
|               |   10.399 |        |    7.804 |        |
+               +----------+--------+----------+--------+
|               |   17.414 |  2.592 |   13.417 |  1.589 |
+---------------+----------+--------+----------+--------+
| make          |   92.756 | 29.790 |   68.732 | 17.462 |
+---------------+----------+--------+----------+--------+
| Total         |  110.170 | 32.382 |   82.149 | 19.051 |
+---------------+----------+--------+----------+--------+
```

http://taskwarrior.org/proj
ews

CMake: Features

- Open-source cross-platform build manager using native tools
  - Visual Studio 6, 7.1, 2005, 2008, 2010, 2012, 2013, 2014
  - Borland make, Nmake, Unix make, MSYS make, MinGW make
  - Ninja
  - Xcode

- IDE Support
  - Code::Blocks
  - CodeLite
  - Eclipse
  - KDevelop
  - Kate
  - Sublime Text

CMake: Features (cont.)

- OSes: HPUX, IRIX, Linux, Mac OSX, QNX, SunOS, Windows, others
- Platform inspection commands can
  - Search for
    - Programs
    - Libraries and Header files
    - Packages
  - Determine hardware specifics like byte order
- Compiler Feature Detection (CMake 3.1)
  - target_compile_features

Why CMake? Quickly adapt to new technologies

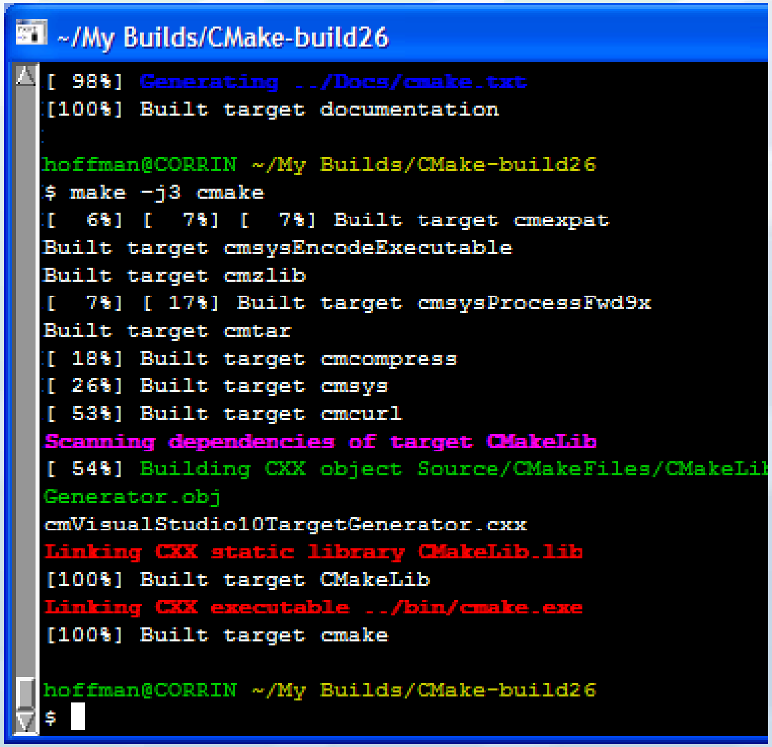
- New build IDE's and compilers
  - Visual Studio releases supported weeks after beta comes out
  - Xcode releases supported weeks after beta comes out
  - ninja (command line build tool from Google) support contributed to CMake as ninja matured
- New compiler support
  - clang
  - gcc versions

CMake: Features (cont.)

- Support for complex custom commands
  - Cuda support
  - Qt’s moc
- Optional component support
- Shared library building (versions for .so supported)
- Create configured .h files
- Single input format for all platforms
- Automatic dependency generation (C, C++, Fortran)
  - Full dependencies: build a target in some directory, and everything this target depends on will be up to date
- Parallel builds
- Out of source builds

CMake: Features (cont.)

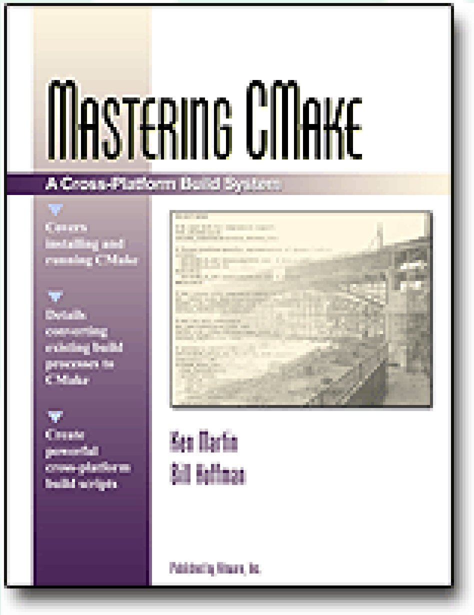
- Color and progress output for make
- Automatically rerun cmake if any cmake input files change (works with Visual Studio using ide macros)
- Graphviz output for visualizing dependency trees
- Full cross platform install() system



~/My Builds/CMake-build26

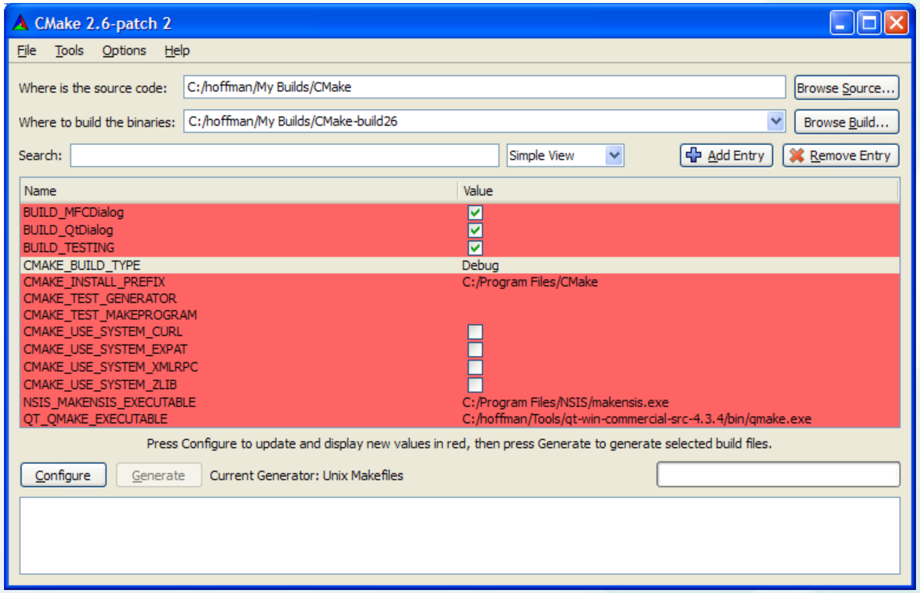
```
[ 98%] Generating ../Docs/cmake.txt
[100%] Built target documentation

hoffman@CORRIN ~/My Builds/CMake-build26
$ make -j3 cmake
[  6%] [  7%] [  7%] Built target cmexpat
Built target cmsysEncodeExecutable
Built target cmzlib
[  7%] [ 17%] Built target cmsysProcessFwd9x
Built target cmtar
[ 18%] Built target cmcompress
[ 26%] Built target cmsys
[ 53%] Built target cmcurl
Scanning dependencies of target CMakeLib
[ 54%] Building CXX object Source/CMakeFiles/CMakeLi
Generator.obj
cmVisualStudio10TargetGenerator.cxx
Linking CXX static library CMakeLib.lib
[100%] Built target CMakeLib
Linking CXX executable ../bin/cmake.exe
[100%] Built target cmake

hoffman@CORRIN ~/My Builds/CMake-build26
$
```

CMake Documentation


Mastering CMake
A Cross-Platform Build System
Covers installing and running CMake
Details converting existing build processes to CMake
Create powerful cross-platform build scripts
Ken Martin
Bill Hoffman
Published by Kitware, Inc.

- Mastering CMake Book
- Web Page: www.cmake.org
- http://www.cmake.org/Wiki/CMake
- mailing list: cmake@cmake.org
  - Full reference documentation
  - http://www.cmake.org/cmake/help/documentation.html
- Ships HTML, man, and command line help
  - Tutorial included and tested in source tree (Tests/Tutorial/)
  - reStructuredText and Sphinx
  - configured files
  - optional build components
  - install rules, test properties
  - system introspection

Cache Editors: cmake-gui (qt), ccmake (curses)



Where is the source code: C:/hoffman/My Builds/CMake

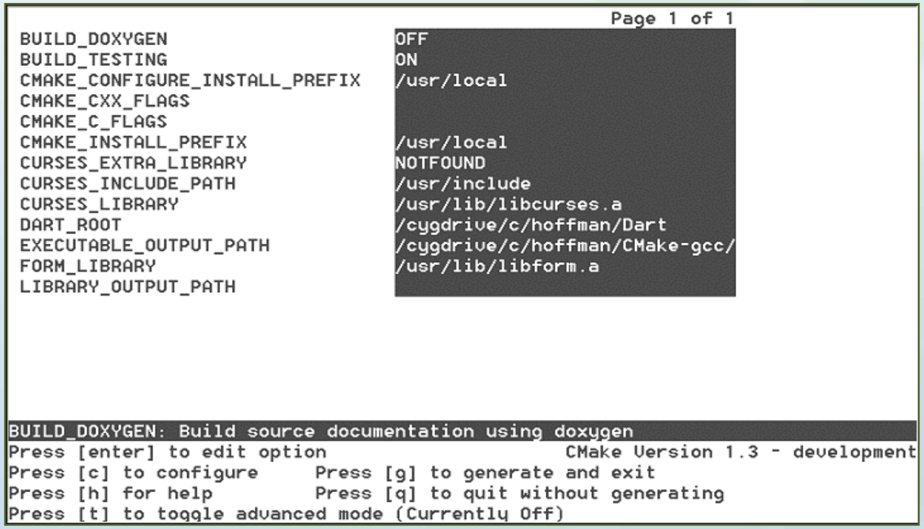
Where to build the binaries: C:/hoffman/My Builds/CMake-build26

Search:

Name	Value
BUILD_MFCDialog	☑
BUILD_QtDialog	☑
BUILD_TESTING	☑
CMAKE_BUILD_TYPE	Debug
CMAKE_INSTALL_PREFIX	C:/Program Files/CMake
CMAKE_TEST_GENERATOR	
CMAKE_TEST_MAKEPROGRAM	
CMAKE_USE_SYSTEM_CURL	□
CMAKE_USE_SYSTEM_EXPAT	□
CMAKE_USE_SYSTEM_XMLRPC	□
CMAKE_USE_SYSTEM_ZLIB	□
NSIS_MAKENSIS_EXECUTABLE	C:/Program Files/NSIS/makensis.exe
QT_QMAKE_EXECUTABLE	C:/hoffman/Tools/qt-win-commercial-src-4.3.4/bin/qmake.exe

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

Current Generator: Unix Makefiles



```
                                                Page 1 of 1
 BUILD_DOXYGEN                    OFF
 BUILD_TESTING                    ON
 CMAKE_CONFIGURE_INSTALL_PREFIX   /usr/local
 CMAKE_CXX_FLAGS
 CMAKE_C_FLAGS
 CMAKE_INSTALL_PREFIX             /usr/local
 CURSES_EXTRA_LIBRARY             NOTFOUND
 CURSES_INCLUDE_PATH              /usr/include
 CURSES_LIBRARY                   /usr/lib/libcurses.a
 DART_ROOT                        /cygdrive/c/hoffman/Dart
 EXECUTABLE_OUTPUT_PATH           /cygdrive/c/hoffman/CMake-gcc/
 FORM_LIBRARY                     /usr/lib/libform.a
 LIBRARY_OUTPUT_PATH







BUILD_DOXYGEN: Build source documentation using doxygen
Press [enter] to edit option                 CMake Version 1.3 - development
Press [c] to configure     Press [g] to generate and exit
Press [h] for help         Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

Running CMake From The Command Line

- Useful for scripted builds or for projects with no options or with options correctly set by default on the first configure

```
#CC=gcc; CXX=g++
#CFLAGS, CXXFLAGS
cd MyProjectSourceDir
mkdir ../MyProjectSourceDir-build
cd ../MyProjectSourceDir-build
cmake ../MyProjectSourceDir

(cmake –Dvar=value)
```

CMake Scripts

- cmake –E command
  - – Cross platform command line utility
  - – Ex. Copy file, Remove file, Compare and conditionally copy, time etc
- cmake –P script.cmake
  - – Cross platform scripting utility
  - – Does not generate cmake_cache
  - – Ignores commands specific to generating build environment

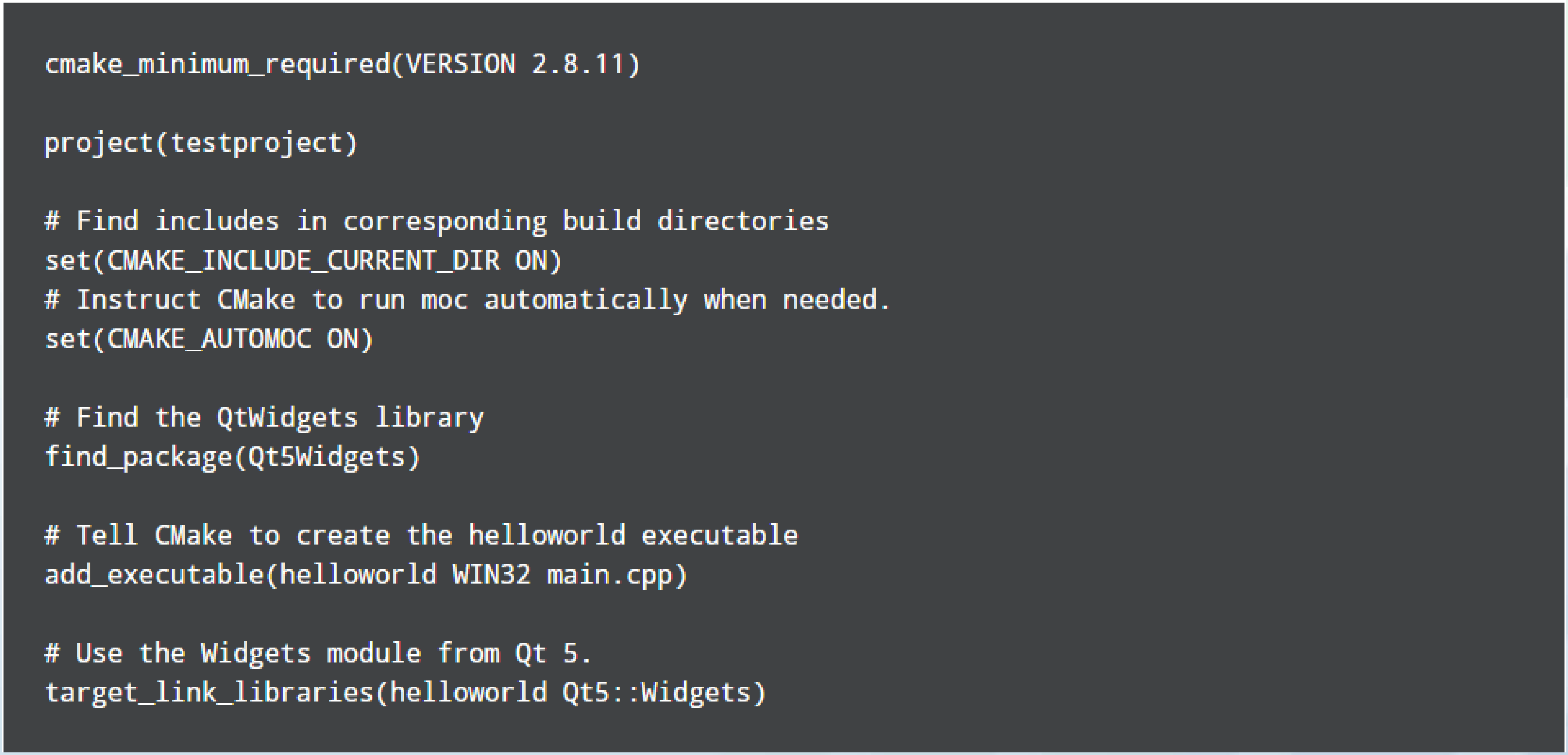
Creating Packages

- Package config files come with installation

```
# myproj-config.cmake
include(${DIR}/myproj-targets.cmake)
set(myproj_INCLUDE_DIRECTORIES
  ${PREFIX}/include/myproj)

# myproj-config-version.cmake
set(PACKAGE_VERSION 1.3)
if("${PACKAGE_FIND_VERSION_MAJOR}" EQUAL 1)
  set(PACKAGE_VERSION_COMPATIBLE 1)
endif()
```

- See “CMakePackageConfigHelpers” module for helper API

Qt5 Ships with CMake config files



```
cmake_minimum_required(VERSION 2.8.11)

project(testproject)

# Find includes in corresponding build directories
set(CMAKE_INCLUDE_CURRENT_DIR ON)
# Instruct CMake to run moc automatically when needed.
set(CMAKE_AUTOMOC ON)

# Find the QtWidgets library
find_package(Qt5Widgets)

# Tell CMake to create the helloworld executable
add_executable(helloworld WIN32 main.cpp)

# Use the Widgets module from Qt 5.
target_link_libraries(helloworld Qt5::Widgets)
```

Usage Requirements – target centric view not directory centric

Propagate

- Include Directories
- Compile Definitions
- Compile Options
- And More!

target_link_libraries and target_include_directories

target_link_libraries - Lets talk different linking types

- PUBLIC
- PRIVATE
- INTERFACE

Target_include_directories - Bring include directories when linking

```
target_include_directories(Foo INTERFACE
                                         ${zlib_dir})
```

- Anything that links to Foo will automatically have the zlib_dir on the include line

target_compile_options and target_compile_definitions

```
target_compile_options(Foo PRIVATE
    -fno-unused-parameter)
```

- Silence really chatty libraries that you can’t modify.

```
target_compile_definitions(Foo PUBLIC
    “NDEBUG”)
```

- Foo will have NDEBUG defined and anything that links to it will have that definition

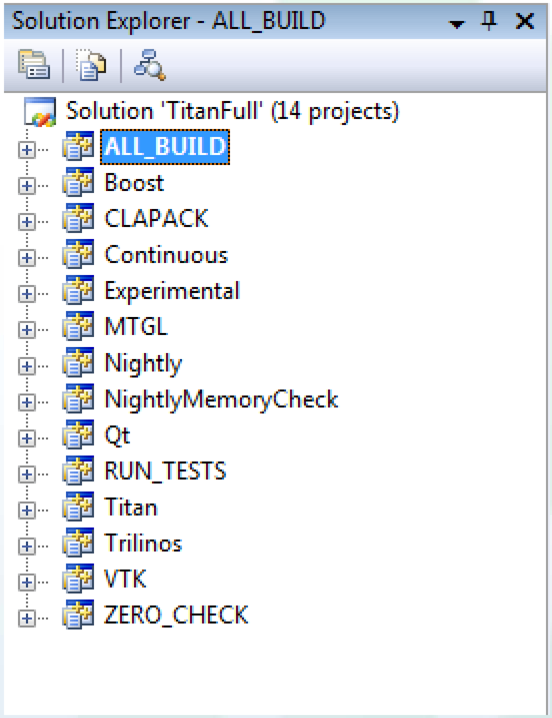
Compiler Feature Detection

```
add_executable(new-app main.cxx)

  target_compile_features(new-app
    cxx_member_templates
    cxx_constexpr
    cxx_generic_lambda
    )

 write_compiler_feature_header(
    FILE "mycompiler_detection.h"
    PREFIX MyPrefix
    FEATURES
    cxx_static_assert
    cxx_final
    cxx_variadic_templates
    )
```

ExternalProject_add

- Module introduced in CMake 2.8
  - Allows the download, configure, build and install of software via custom commands
- Kitware Source Article: October 2009
  - http://www.kitware.com/products/html/BuildingExternalProjectsWithCMake2.8.html
- Funded by Army Research Lab (ARL) – Computational Science Environment (CSE)

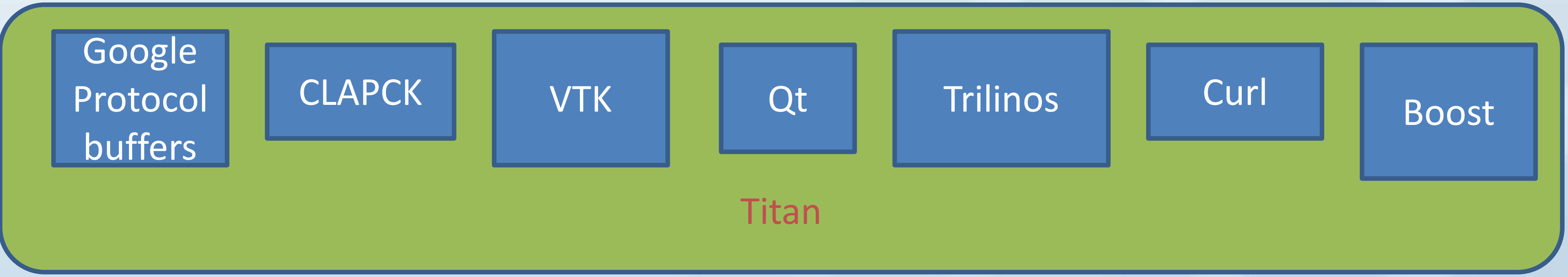
Titan Example



Solution Explorer - ALL_BUILD

Solution 'TitanFull' (14 projects)

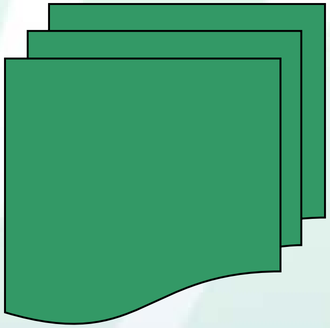
- ALL_BUILD
- Boost
- CLAPACK
- Continuous
- Experimental
- MTGL
- Nightly
- NightlyMemoryCheck
- Qt
- RUN_TESTS
- Titan
- Trilinos
- VTK
- ZERO_CHECK



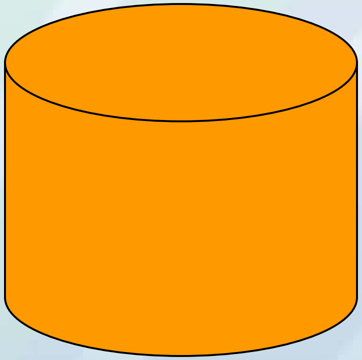
Test Data Management ExternalData

Distributed Version Control

- Meant for source code, not data
- Local history of *source* is good
  - Often modified → interesting history
  - Line-wise commits → good deltas
  - Fast **`log`**, **`blame`**, etc.

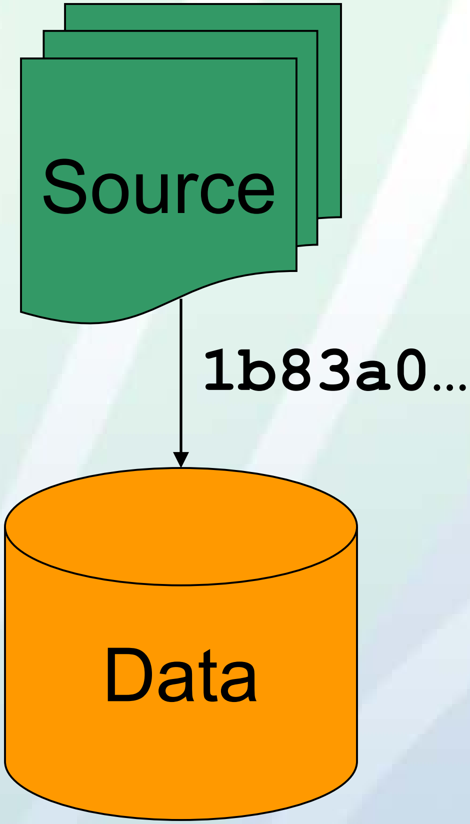


- Local history of *data* is bad
  - Rarely modified → boring history
  - Whole-file commits → poor deltas
  - No **`blame`** for binary files



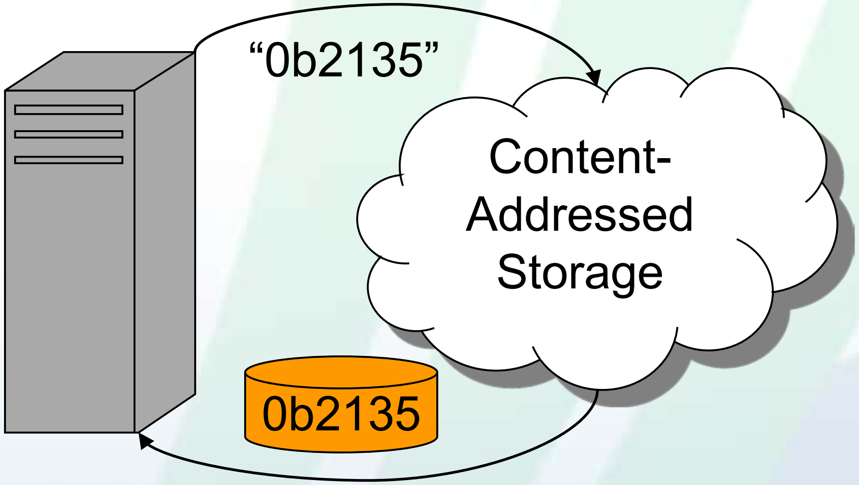
Separating Data from Source

- Source must reference data
- Tests need matching data
- Links must be *unambiguous*
- Links must be *lightweight*
- Answer: *content hash*



Content-Addressed Storage

- Arbitrary locations
  - – Local machine
  - – Private server
  - – Internet server
- Content verified by hash
- No need to trust provider if hash is strong



ExternalData Module - Source

- Start with real data file in source tree (locally)
- Source code references data by original file name

```
$ cat CMakeLists.txt
ExternalData_add_test(ITKData
  NAME CellularSegmentation2Test
  COMMAND SegmentationExamples9 CellularSegmentation2Test
                    DATA{../Data/BrainWeb/brainweb1e1a10f20.mha}
  ...)
```

- Test works with real data file out of the box
- Then replace data file by a “content link”

```
$ cat ../Data/BrainWeb/brainweb1e1a10f20.mha.md5
0b2135e2035e5bd84d82f4929e68fbdc
```

- Conversion to content link can be scripted
- Data go to local or remote content-addressed storage

ExternalData Module - Build

- Build system handles creation of local instance
- Fetches data from arbitrary content-addressed storage

```
$ make ITKData
Generating ExternalData/Examples/Data/BrainWeb/brainweb1e1a10f20.mha
-- Fetching "http://.../MD5/0b2135e2035e5bd84d82f4929e68fbdc"
-- [download 100% complete]
-- Downloaded object: "ExternalData/Objects/MD5/0b2135e2035e5bd84d82f4929e68fbdc"
```

- Test uses local instance by original file name

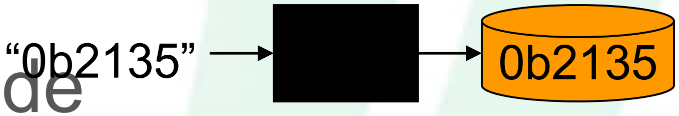
```
$ bin/SegmentationExamples9 CellularSegmentation2Test \
   ExternalData/Examples/Data/BrainWeb/brainweb1e1a10f20.mha ...
```

- Original file name provided by symbolic link if possible

```
$ readlink ExternalData/Examples/Data/BrainWeb/brainweb1e1a10f20.mha
../../../Objects/MD5/0b2135e2035e5bd84d82f4929e68fbdc
```

ExternalData Module - Fetch

- Method is a black box
  - – Hidden from source code
  - – Can change in future without breaking old versions



- Configured list of URL templates
  - – `file:///local/%(algo)/%(hash)`
  - – `http://server.local/%(algo)/%(hash)`
  - – `http://midas.kitware.com/...?algorithm=%(algo)&hash=%(hash)`
- Try each location in order
  - – Substitute for %(algo) and %(hash) in URL
  - – Download and check content hash
  - – Done if hash matches, else continue

Coming to CMake

- Android.mk generation for CMake built libraries
- Integrated Android platform support simplifying tool chain file

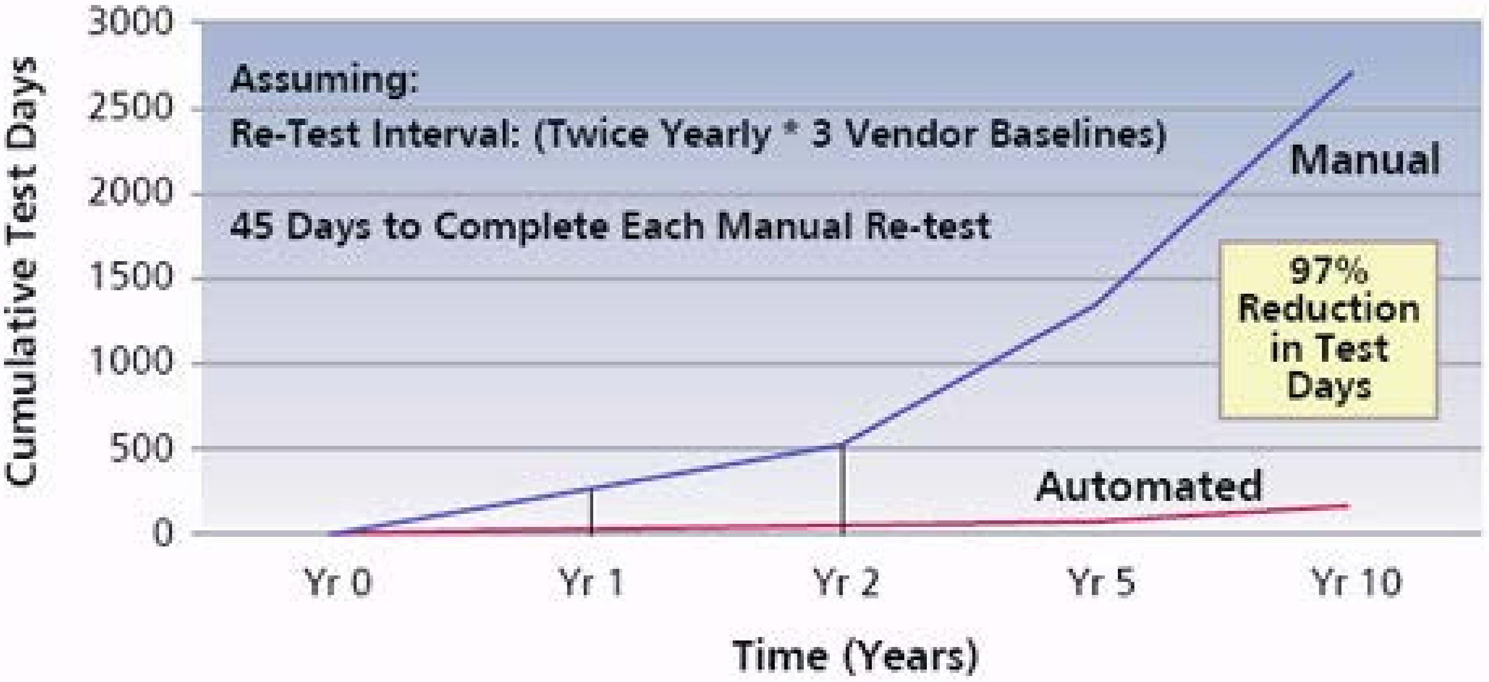
Simple Qt Example

```
cmake_minimum_required(VERSION 2.8)
project(helloQt)
# find required dependencies
find_package(Qt4 REQUIRED)
# create the executable
add_executable(helloQt WIN32 MACOSX_BUNDLE myqt.cxx )
target_link_libraries(helloQt ${QT_QTMAIN_LIBRARY} ${QT_LIBRARIES})
# installation and packaging
install(TARGETS helloQt DESTINATION bin)
include (InstallRequiredSystemLibraries)
set (CPACK_PACKAGE_VERSION_MAJOR "1")
set (CPACK_PACKAGE_VERSION_MINOR "0")
set(CPACK_PACKAGE_EXECUTABLES "helloQt" "Hello Qt")
include (CPack)
```

Simple Qt Example with Boost

```
cmake_minimum_required(VERSION 2.8)
project(helloQt)
# find required dependencies
find_package(Qt4 REQUIRED)
include(${QT_USE_FILE})
set( Boost_USE_STATIC_LIBS ON )
find_package(Boost REQUIRED signals)
include_directories(${Boost_INCLUDE_DIRS})
# create the executable
add_executable(helloQt WIN32 MACOSX_BUNDLE myqt.cxx )
target_link_libraries(helloQt ${QT_QTMAIN_LIBRARY} ${QT_LIBRARIES}
  ${Boost_LIBRARIES}   )
# installation and packaging
install(TARGETS helloQt DESTINATION bin)
include (InstallRequiredSystemLibraries)
set (CPACK_PACKAGE_VERSION_MAJOR "1")
set (CPACK_PACKAGE_VERSION_MINOR "0")
set(CPACK_PACKAGE_EXECUTABLES "helloQt" "Hello Qt")
include (CPack)
```

Automatic Testing Benefits



Assuming:
Re-Test Interval: (Twice Yearly * 3 Vendor Baselines)

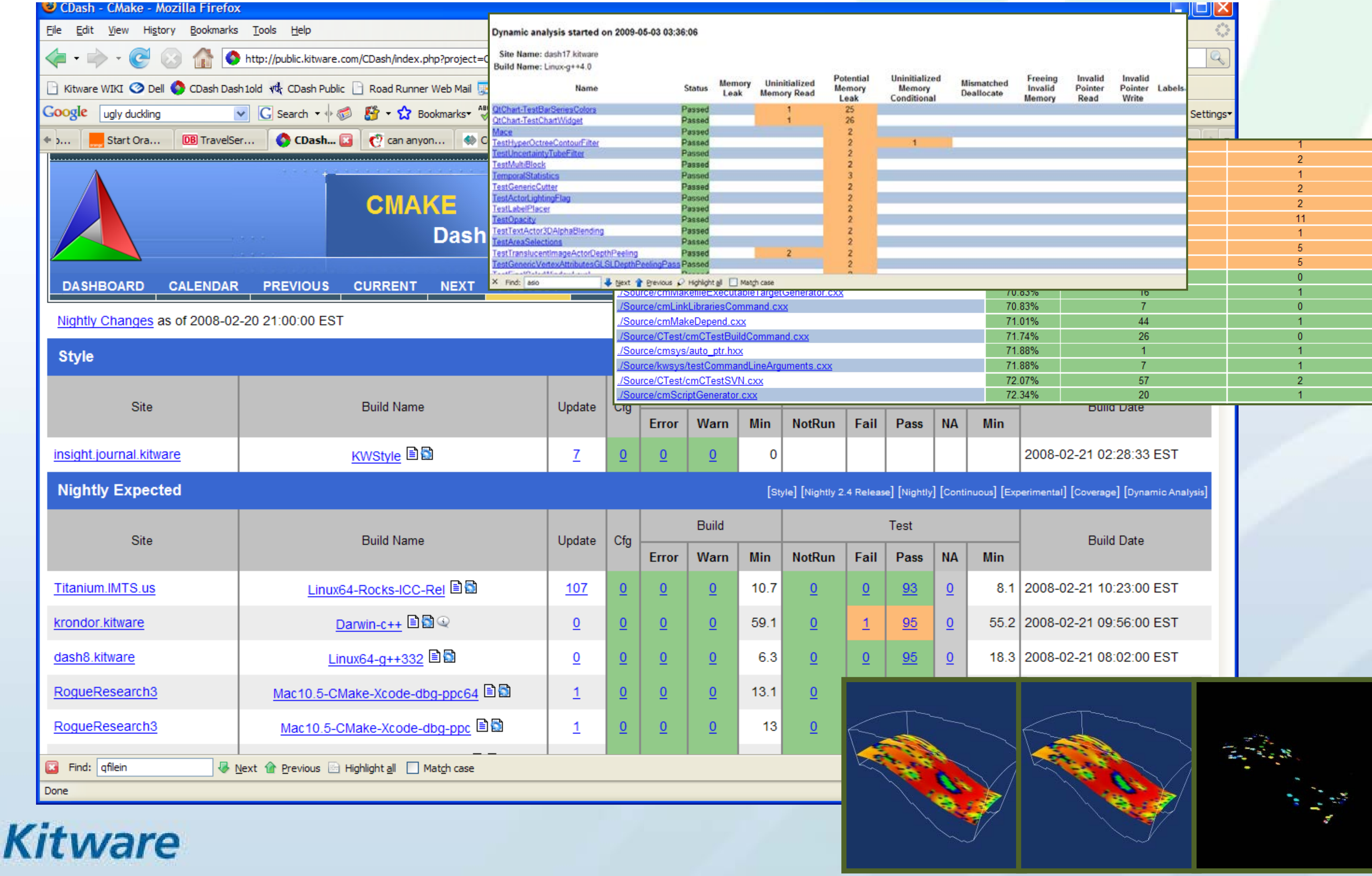
45 Days to Complete Each Manual Re-test

97% Reduction in Test Days

Figure 2. "Example Automated Software Testing Savings over Time"

"Automated Software Testing,"
1999, Dustin, et al, Addison Wesley

CDash Dashboard www.cdash.org



Dynamic analysis started on 2009-05-03 03:36:06

Site Name: dash17.kitware
Build Name: Linux-g++4.0

Name	Status	Memory Leak	Uninitialized Memory Read	Potential Memory Leak	Uninitialized Memory Conditional	Mismatched Deallocate	Freeing Invalid Memory	Invalid Pointer Read	Invalid Pointer Write	Labels
QtChart-TestBarSeriesColors	Passed		1	25						
QtChart-TestChartWidget	Passed		1	26						
Mace	Passed			2						
TestHyperOctreeContourFilter	Passed			2	1					
TestUncertaintyTubeFilter	Passed			2						
TestMultiBlock	Passed			2						
TemporalStatistics	Passed			3						
TestGenericCutter	Passed			2						
TestActorLightingFlag	Passed			2						
TestLabelPlacer	Passed			2						
TestOpacity	Passed			2						
TestTextActor3DAlphaBlending	Passed			2						
TestAreaSelections	Passed			2						
TestTranslucentImageActorDepthPeeling	Passed		2	2						
TestGenericVertexAttributesGLSLDepthPeelingPass	Passed			2						

Find: asio Next Previous Highlight all ☐ Match case

			1
			2
			1
			2
			2
			11
			1
			5
			5
			0
./Source/cmMakefileExecutableTargetGenerator.cxx	70.83%	16	1
./Source/cmLinkLibrariesCommand.cxx	70.83%	7	0
./Source/cmMakeDepend.cxx	71.01%	44	1
./Source/CTest/cmCTestBuildCommand.cxx	71.74%	26	0
./Source/cmsys/auto_ptr.hxx	71.88%	1	1
./Source/kwsys/testCommandLineArguments.cxx	71.88%	7	1
./Source/CTest/cmCTestSVN.cxx	72.07%	57	2
./Source/cmScriptGenerator.cxx	72.34%	20	1

Nightly Changes as of 2008-02-20 21:00:00 EST

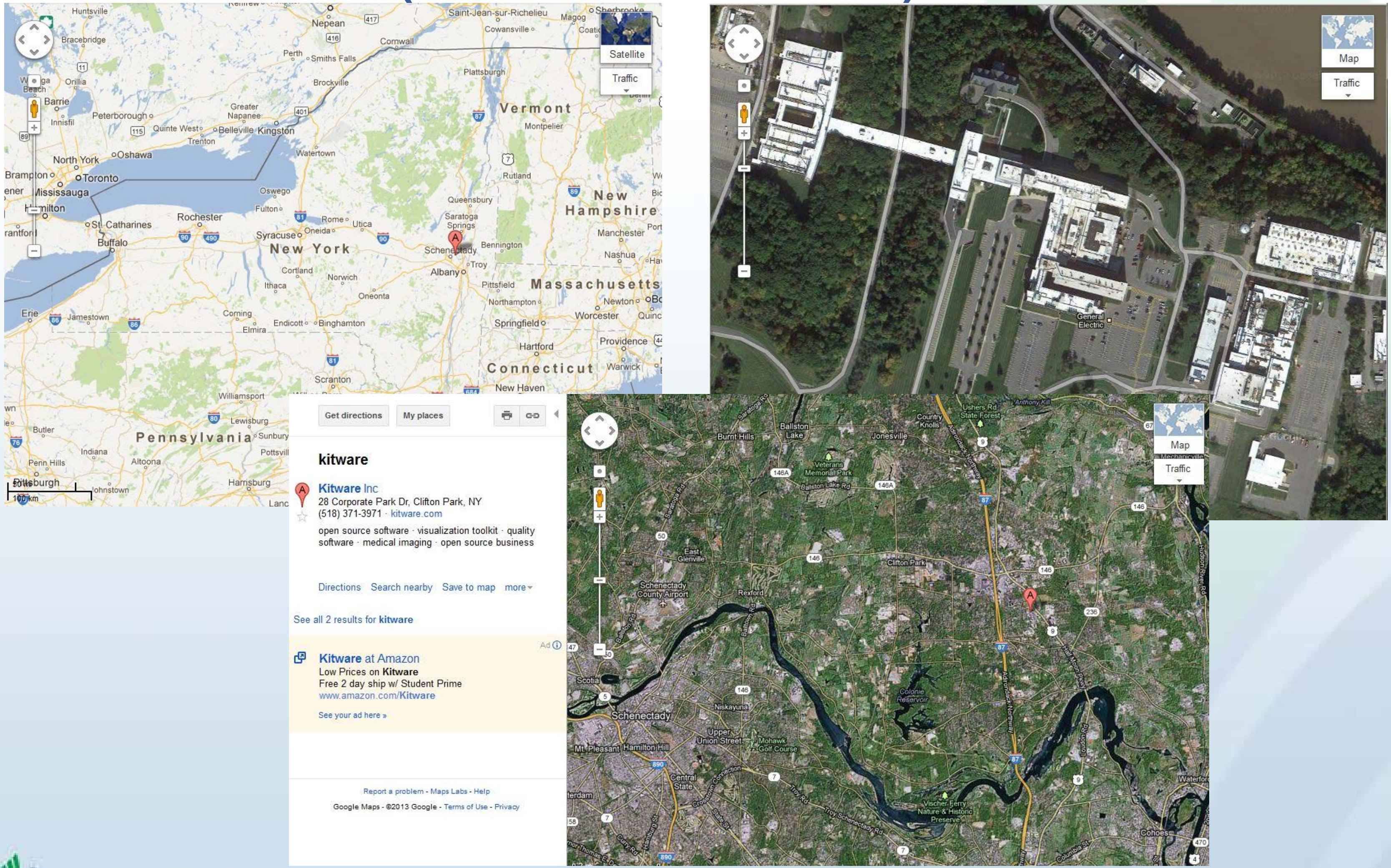
Style

Site	Build Name	Update	Cfg	Build			Test					Build Date
				Error	Warn	Min	NotRun	Fail	Pass	NA	Min	
insight.journal.kitware	KWStyle	7	0	0	0	0						2008-02-21 02:28:33 EST

Nightly Expected [Style] [Nightly 2.4 Release] [Nightly] [Continuous] [Experimental] [Coverage] [Dynamic Analysis]

Site	Build Name	Update	Cfg	Build			Test					Build Date
				Error	Warn	Min	NotRun	Fail	Pass	NA	Min	
Titanium.IMTS.us	Linux64-Rocks-ICC-Rel	107	0	0	0	10.7	0	0	93	0	8.1	2008-02-21 10:23:00 EST
krondor.kitware	Darwin-c++	0	0	0	0	59.1	0	1	95	0	55.2	2008-02-21 09:56:00 EST
dash8.kitware	Linux64-g++332	0	0	0	0	6.3	0	0	95	0	18.3	2008-02-21 08:02:00 EST
RogueResearch3	Mac10.5-CMake-Xcode-dbg-ppc64	1	0	0	0	13.1	0					
RogueResearch3	Mac10.5-CMake-Xcode-dbg-ppc	1	0	0	0	13	0					

Six Sigma and Quality Research Software (GE Research)



Get directions My places

kitware

Kitware Inc
28 Corporate Park Dr, Clifton Park, NY
(518) 371-3971 · kitware.com
open source software · visualization toolkit · quality software · medical imaging · open source business

Directions Search nearby Save to map more

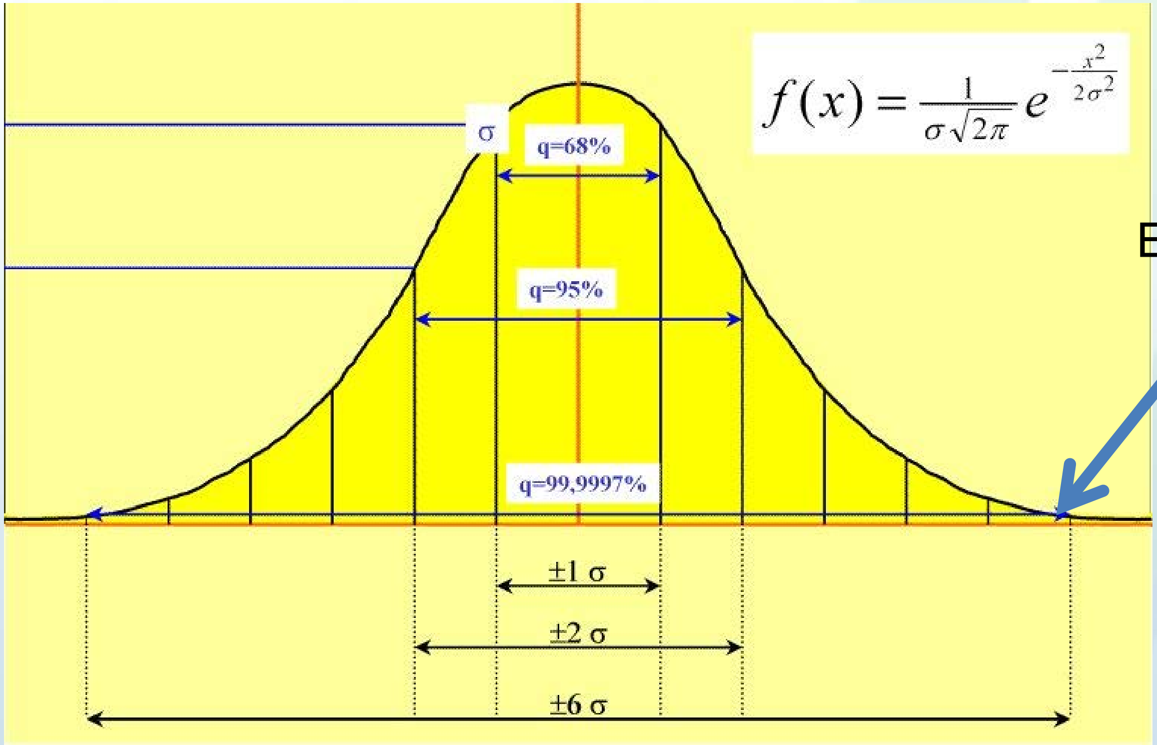
See all 2 results for kitware

Ad

Kitware at Amazon
Low Prices on Kitware
Free 2 day ship w/ Student Prime
www.amazon.com/Kitware
See your ad here »

Report a problem - Maps Labs - Help
Google Maps - ©2013 Google - Terms of Use - Privacy

Six Sigma and Quality Research Software



Errors / Defects

CDash can be used with other tools

- Jenkins – Project Tango and others
- Buildbot
  - Extended buildbot to run the configure, build, and test steps for CMake-based projects with support for uploading build artifacts (packages, logs, etc.)
  - Buildbot parses the output of CTest to determine error and warning status
- Travis
- CircleCI
- Simple ctest cronjobs

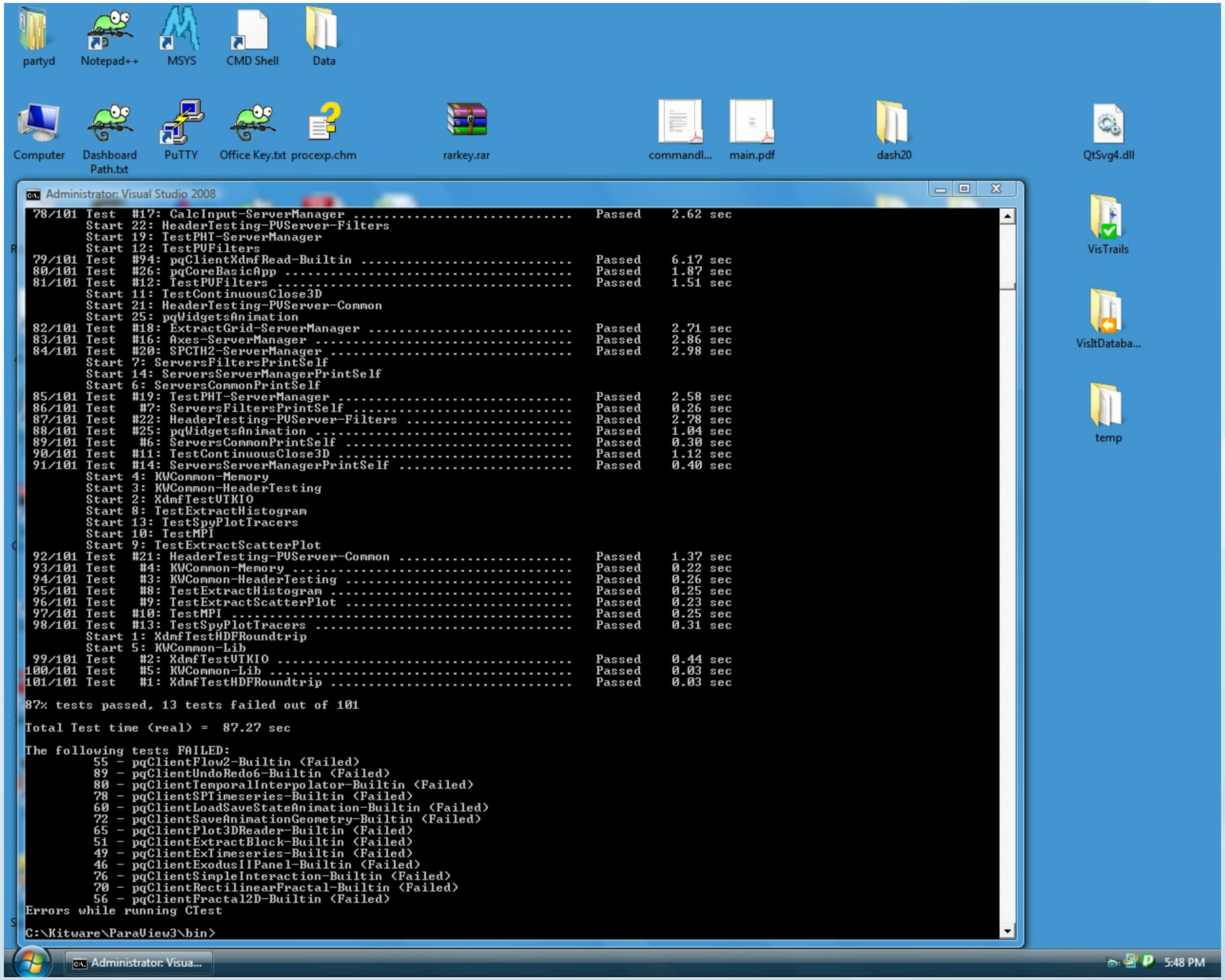
CDash github/gitlab integration

- Buildbot is used to act on actions based on GitLab comments to trigger builds
- Builds may be constrained to specific builders or machines for targeted testing
- Buildbot reports back to merge requests once all the relevant builders have completed with links to buildbot itself

Testing with CMake

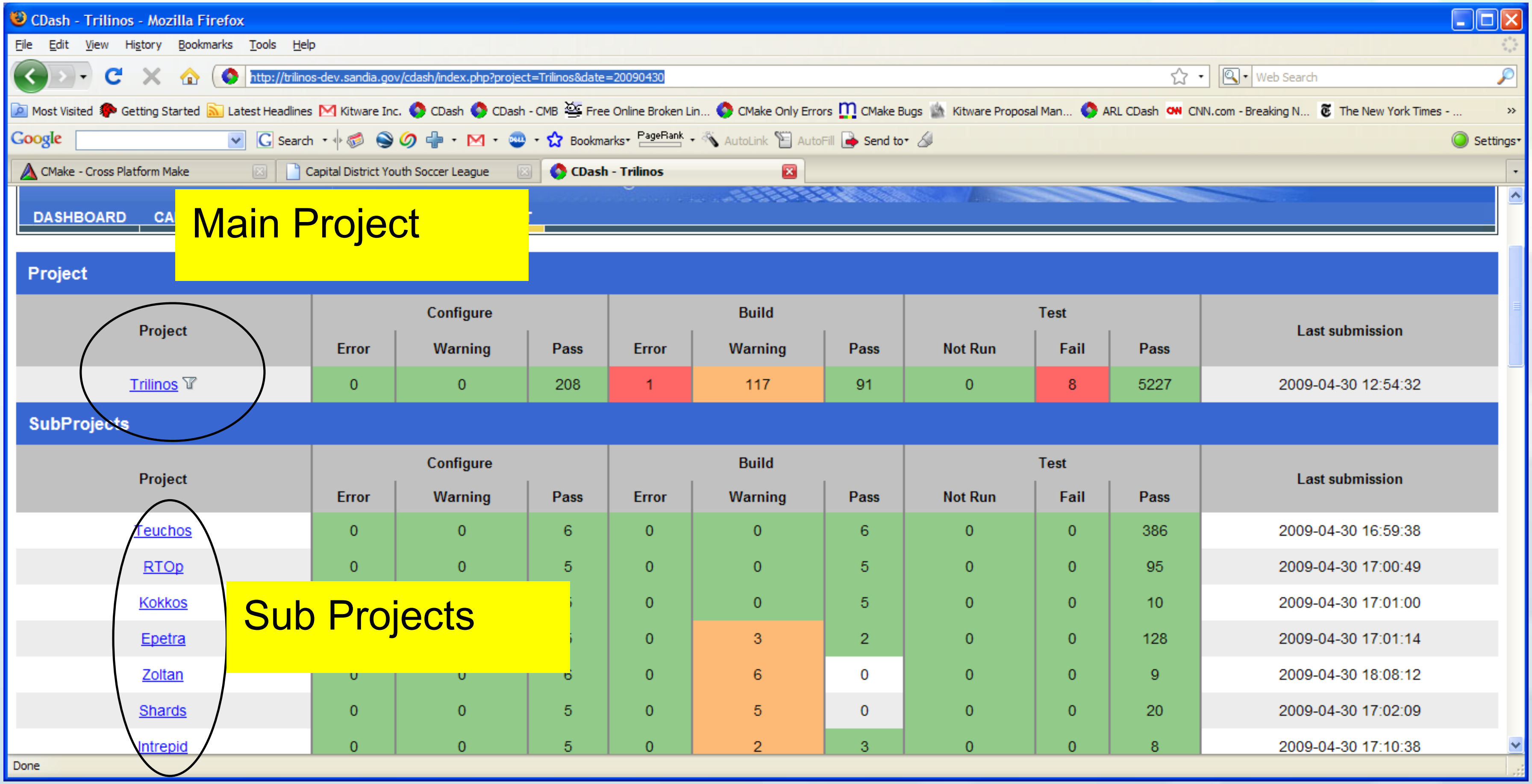
- include(CTest) or enable_testing() – enables the ADD_TEST commands in a project
- add_test (testname exename arg1 arg2   )
  - Executable should return 0 for a test that passes
- ctest – an executable that is distributed with cmake that can run tests in a project.
- CDash testing – Create web based dashboards to show the results of tests for each day

Video of ParaView Nightly Testing



```
 78/101 Test  #17: CalcInput-ServerManager .............................   Passed    2.62 sec
        Start 22: HeaderTesting-PVServer-Filters
        Start 19: TestPHT-ServerManager
        Start 12: TestPVFilters
 79/101 Test  #94: pqClientXdmfRead-Builtin ............................   Passed    6.17 sec
 80/101 Test  #26: pqCoreBasicApp ......................................   Passed    1.87 sec
 81/101 Test  #12: TestPVFilters .......................................   Passed    1.51 sec
        Start 11: TestContinuousClose3D
        Start 21: HeaderTesting-PVServer-Common
        Start 25: pqWidgetsAnimation
 82/101 Test  #18: ExtractGrid-ServerManager ...........................   Passed    2.71 sec
 83/101 Test  #16: Axes-ServerManager ..................................   Passed    2.86 sec
 84/101 Test  #20: SPCTH2-ServerManager ................................   Passed    2.98 sec
        Start 7: ServersFiltersPrintSelf
        Start 14: ServersServerManagerPrintSelf
        Start 6: ServersCommonPrintSelf
 85/101 Test  #19: TestPHT-ServerManager ...............................   Passed    2.58 sec
 86/101 Test   #7: ServersFiltersPrintSelf .............................   Passed    0.26 sec
 87/101 Test  #22: HeaderTesting-PVServer-Filters ......................   Passed    2.78 sec
 88/101 Test  #25: pqWidgetsAnimation ..................................   Passed    1.04 sec
 89/101 Test   #6: ServersCommonPrintSelf ..............................   Passed    0.30 sec
 90/101 Test  #11: TestContinuousClose3D ...............................   Passed    1.12 sec
 91/101 Test  #14: ServersServerManagerPrintSelf .......................   Passed    0.40 sec
        Start 4: KWCommon-Memory
        Start 3: KWCommon-HeaderTesting
        Start 2: XdmfTestVTKIO
        Start 8: TestExtractHistogram
        Start 13: TestSpyPlotTracers
        Start 10: TestMPI
        Start 9: TestExtractScatterPlot
 92/101 Test  #21: HeaderTesting-PVServer-Common .......................   Passed    1.37 sec
 93/101 Test   #4: KWCommon-Memory .....................................   Passed    0.22 sec
 94/101 Test   #3: KWCommon-HeaderTesting ..............................   Passed    0.26 sec
 95/101 Test   #8: TestExtractHistogram ................................   Passed    0.25 sec
 96/101 Test   #9: TestExtractScatterPlot ..............................   Passed    0.23 sec
 97/101 Test  #10: TestMPI .............................................   Passed    0.25 sec
 98/101 Test  #13: TestSpyPlotTracers ..................................   Passed    0.31 sec
        Start 1: XdmfTestHDFRoundtrip
        Start 5: KWCommon-Lib
 99/101 Test   #2: XdmfTestVTKIO .......................................   Passed    0.44 sec
100/101 Test   #5: KWCommon-Lib ........................................   Passed    0.03 sec
101/101 Test   #1: XdmfTestHDFRoundtrip ................................   Passed    0.03 sec

87% tests passed, 13 tests failed out of 101

Total Test time (real) =  87.27 sec

The following tests FAILED:
          55 - pqClientFlow2-Builtin (Failed)
          89 - pqClientUndoRedo6-Builtin (Failed)
          80 - pqClientTemporalInterpolator-Builtin (Failed)
          78 - pqClientSPTimeseries-Builtin (Failed)
          60 - pqClientLoadSaveStateAnimation-Builtin (Failed)
          72 - pqClientSaveAnimationGeometry-Builtin (Failed)
          65 - pqClientPlot3DReader-Builtin (Failed)
          51 - pqClientExtractBlock-Builtin (Failed)
          49 - pqClientExTimeseries-Builtin (Failed)
          46 - pqClientExodusIIPanel-Builtin (Failed)
          76 - pqClientSimpleInteraction-Builtin (Failed)
          70 - pqClientRectilinearFractal-Builtin (Failed)
          56 - pqClientFractal2D-Builtin (Failed)
Errors while running CTest

C:\Kitware\ParaView3\bin>
```

Trilinos (Multi-Package Dashboard) http://trilinos-dev.sandia.gov/cdash/index.php


CDash - Trilinos - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://trilinos-dev.sandia.gov/cdash/index.php?project=Trilinos&date=20090430
Web Search
Most Visited Getting Started Latest Headlines Kitware Inc. CDash CDash - CMB Free Online Broken Lin... CMake Only Errors CMake Bugs Kitware Proposal Man... ARL CDash CNN.com - Breaking N... The New York Times - ...
Google Search Bookmarks PageRank AutoLink AutoFill Send to Settings
CMake - Cross Platform Make
Capital District Youth Soccer League
CDash - Trilinos
DASHBOARD
CA
Done

Main Project

Project

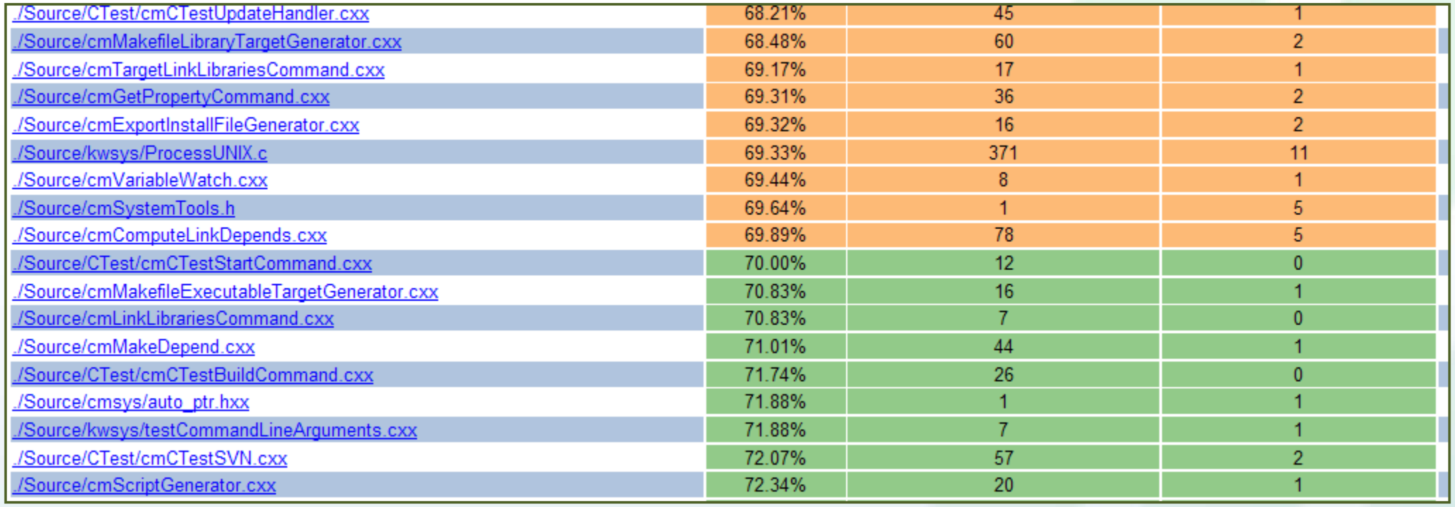
Project	Configure			Build			Test			Last submission
	Error	Warning	Pass	Error	Warning	Pass	Not Run	Fail	Pass	
Trilinos	0	0	208	1	117	91	0	8	5227	2009-04-30 12:54:32

SubProjects

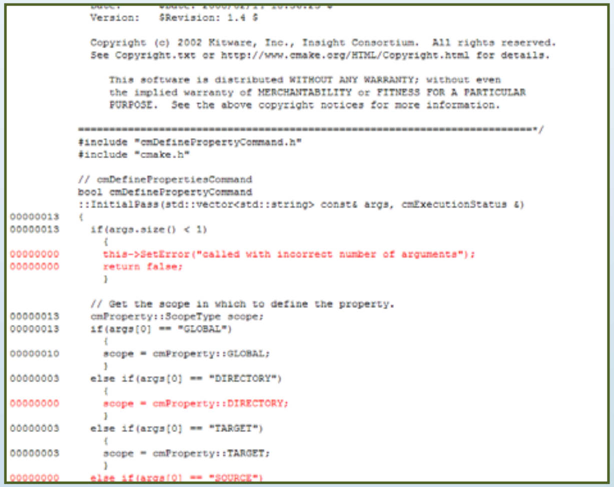
Project	Configure			Build			Test			Last submission
	Error	Warning	Pass	Error	Warning	Pass	Not Run	Fail	Pass	
Teuchos	0	0	6	0	0	6	0	0	386	2009-04-30 16:59:38
RTOp	0	0	5	0	0	5	0	0	95	2009-04-30 17:00:49
Kokkos	[illegible]	[illegible]	5	0	0	5	0	0	10	2009-04-30 17:01:00
Epetra	[illegible]	[illegible]	5	0	3	2	0	0	128	2009-04-30 17:01:14
Zoltan	0	0	6	0	6	0	0	0	9	2009-04-30 18:08:12
Shards	0	0	5	0	5	0	0	0	20	2009-04-30 17:02:09
Intrepid	0	0	5	0	2	3	0	0	8	2009-04-30 17:10:38

Sub Projects

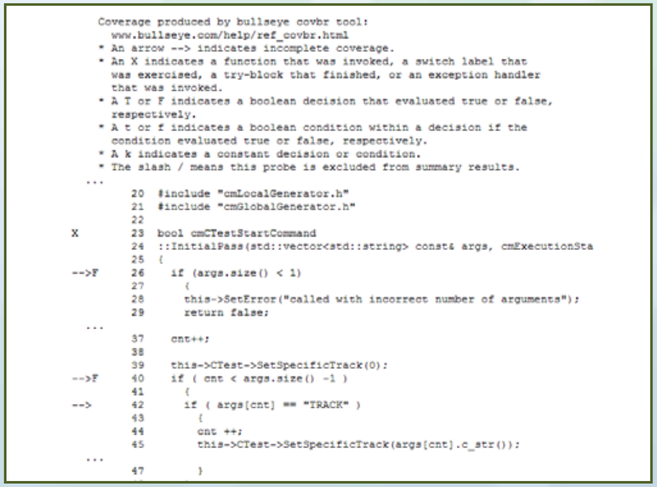
Coverage Display GCov/Bullseye



./Source/CTest/cmCTestUpdateHandler.cxx	68.21%	45	1
./Source/cmMakefileLibraryTargetGenerator.cxx	68.48%	60	2
./Source/cmTargetLinkLibrariesCommand.cxx	69.17%	17	1
./Source/cmGetPropertyCommand.cxx	69.31%	36	2
./Source/cmExportInstallFileGenerator.cxx	69.32%	16	2
./Source/kwsys/ProcessUNIX.c	69.33%	371	11
./Source/cmVariableWatch.cxx	69.44%	8	1
./Source/cmSystemTools.h	69.64%	1	5
./Source/cmComputeLinkDepends.cxx	69.89%	78	5
./Source/CTest/cmCTestStartCommand.cxx	70.00%	12	0
./Source/cmMakefileExecutableTargetGenerator.cxx	70.83%	16	1
./Source/cmLinkLibrariesCommand.cxx	70.83%	7	0
./Source/cmMakeDepend.cxx	71.01%	44	1
./Source/CTest/cmCTestBuildCommand.cxx	71.74%	26	0
./Source/cmsys/auto_ptr.hxx	71.88%	1	1
./Source/kwsys/testCommandLineArguments.cxx	71.88%	7	1
./Source/CTest/cmCTestSVN.cxx	72.07%	57	2
./Source/cmScriptGenerator.cxx	72.34%	20	1

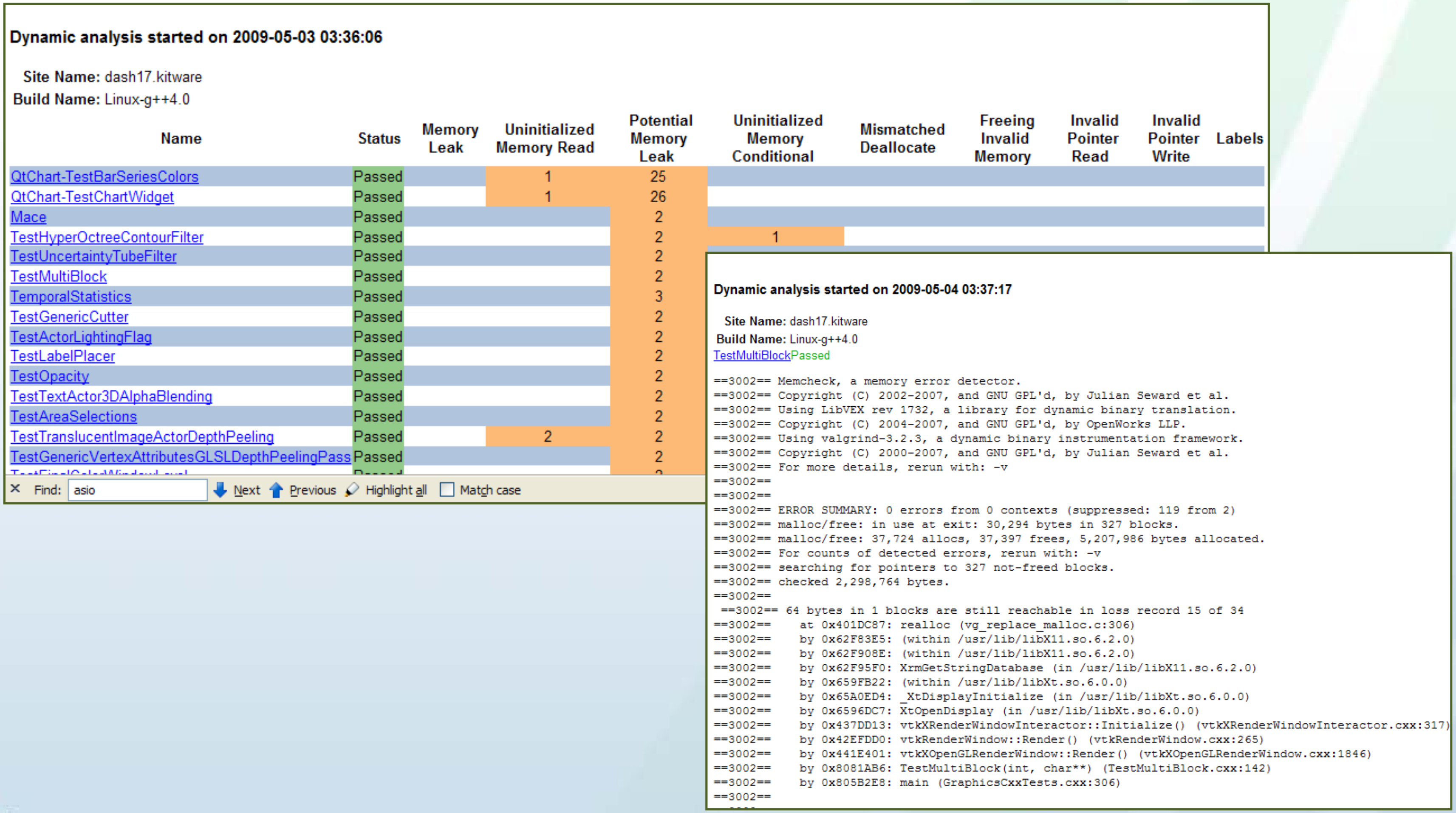


```
               Version:   $Revision: 1.4 $

               Copyright (c) 2002 Kitware, Inc., Insight Consortium.  All rights reserved.
               See Copyright.txt or http://www.cmake.org/HTML/Copyright.html for details.

                  This software is distributed WITHOUT ANY WARRANTY; without even
                  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
                  PURPOSE.  See the above copyright notices for more information.

             =========================================================================*/
             #include "cmDefinePropertyCommand.h"
             #include "cmake.h"

             // cmDefinePropertiesCommand
             bool cmDefinePropertyCommand
             ::InitialPass(std::vector<std::string> const& args, cmExecutionStatus &)
00000013     {
00000013       if(args.size() < 1)
                 {
00000000         this->SetError("called with incorrect number of arguments");
00000000         return false;
                 }

               // Get the scope in which to define the property.
00000013       cmProperty::ScopeType scope;
00000013       if(args[0] == "GLOBAL")
                 {
00000010         scope = cmProperty::GLOBAL;
                 }
00000003       else if(args[0] == "DIRECTORY")
                 {
00000000         scope = cmProperty::DIRECTORY;
                 }
00000003       else if(args[0] == "TARGET")
                 {
00000003         scope = cmProperty::TARGET;
                 }
00000000       else if(args[0] == "SOURCE")
```



```
          Coverage produced by bullseye covbr tool:
            www.bullseye.com/help/ref_covbr.html
          * An arrow --> indicates incomplete coverage.
          * An X indicates a function that was invoked, a switch label that
            was exercised, a try-block that finished, or an exception handler
            that was invoked.
          * A T or F indicates a boolean decision that evaluated true or false,
            respectively.
          * A t or f indicates a boolean condition within a decision if the
            condition evaluated true or false, respectively.
          * A k indicates a constant decision or condition.
          * The slash / means this probe is excluded from summary results.
        ...
             20  #include "cmLocalGenerator.h"
             21  #include "cmGlobalGenerator.h"
             22
X            23  bool cmCTestStartCommand
             24  ::InitialPass(std::vector<std::string> const& args, cmExecutionSta
             25  {
-->F         26    if (args.size() < 1)
             27      {
             28      this->SetError("called with incorrect number of arguments");
             29      return false;
        ...
             37    cnt++;
             38
             39    this->CTest->SetSpecificTrack(0):
-->F         40    if ( cnt < args.size() -1 )
             41      {
-->          42      if ( args[cnt] == "TRACK" )
             43        {
             44        cnt ++;
             45        this->CTest->SetSpecificTrack(args[cnt].c_str()):
        ...
             47        }
```

Valgrind / Purify



Dynamic analysis started on 2009-05-03 03:36:06

Site Name: dash17.kitware
Build Name: Linux-g++4.0

Name	Status	Memory Leak	Uninitialized Memory Read	Potential Memory Leak	Uninitialized Memory Conditional	Mismatched Deallocate	Freeing Invalid Memory	Invalid Pointer Read	Invalid Pointer Write	Labels
QtChart-TestBarSeriesColors	Passed		1	25						
QtChart-TestChartWidget	Passed		1	26						
Mace	Passed			2						
TestHyperOctreeContourFilter	Passed			2	1					
TestUncertaintyTubeFilter	Passed			2						
TestMultiBlock	Passed			2						
TemporalStatistics	Passed			3						
TestGenericCutter	Passed			2						
TestActorLightingFlag	Passed			2						
TestLabelPlacer	Passed			2						
TestOpacity	Passed			2						
TestTextActor3DAlphaBlending	Passed			2						
TestAreaSelections	Passed			2						
TestTranslucentImageActorDepthPeeling	Passed		2	2						
TestGenericVertexAttributesGLSLDepthPeelingPass	Passed			2						

Dynamic analysis started on 2009-05-04 03:37:17

Site Name: dash17.kitware
Build Name: Linux-g++4.0
TestMultiBlockPassed

```
==3002== Memcheck, a memory error detector.
==3002== Copyright (C) 2002-2007, and GNU GPL'd, by Julian Seward et al.
==3002== Using LibVEX rev 1732, a library for dynamic binary translation.
==3002== Copyright (C) 2004-2007, and GNU GPL'd, by OpenWorks LLP.
==3002== Using valgrind-3.2.3, a dynamic binary instrumentation framework.
==3002== Copyright (C) 2000-2007, and GNU GPL'd, by Julian Seward et al.
==3002== For more details, rerun with: -v
==3002==
==3002==
==3002== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 119 from 2)
==3002== malloc/free: in use at exit: 30,294 bytes in 327 blocks.
==3002== malloc/free: 37,724 allocs, 37,397 frees, 5,207,986 bytes allocated.
==3002== For counts of detected errors, rerun with: -v
==3002== searching for pointers to 327 not-freed blocks.
==3002== checked 2,298,764 bytes.
==3002==
 ==3002== 64 bytes in 1 blocks are still reachable in loss record 15 of 34
==3002==    at 0x401DC87: realloc (vg_replace_malloc.c:306)
==3002==    by 0x62F83E5: (within /usr/lib/libX11.so.6.2.0)
==3002==    by 0x62F908E: (within /usr/lib/libX11.so.6.2.0)
==3002==    by 0x62F95F0: XrmGetStringDatabase (in /usr/lib/libX11.so.6.2.0)
==3002==    by 0x659FB22: (within /usr/lib/libXt.so.6.0.0)
==3002==    by 0x65A0ED4: _XtDisplayInitialize (in /usr/lib/libXt.so.6.0.0)
==3002==    by 0x6596DC7: XtOpenDisplay (in /usr/lib/libXt.so.6.0.0)
==3002==    by 0x437DD13: vtkXRenderWindowInteractor::Initialize() (vtkXRenderWindowInteractor.cxx:317)
==3002==    by 0x42EFDD0: vtkRenderWindow::Render() (vtkRenderWindow.cxx:265)
==3002==    by 0x441E401: vtkXOpenGLRenderWindow::Render() (vtkXOpenGLRenderWindow.cxx:1846)
==3002==    by 0x8081AB6: TestMultiBlock(int, char**) (TestMultiBlock.cxx:142)
==3002==    by 0x805B2E8: main (GraphicsCxxTests.cxx:306)
==3002==
```

CDash Email Notification

A submission to CDash for the project CMake 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.

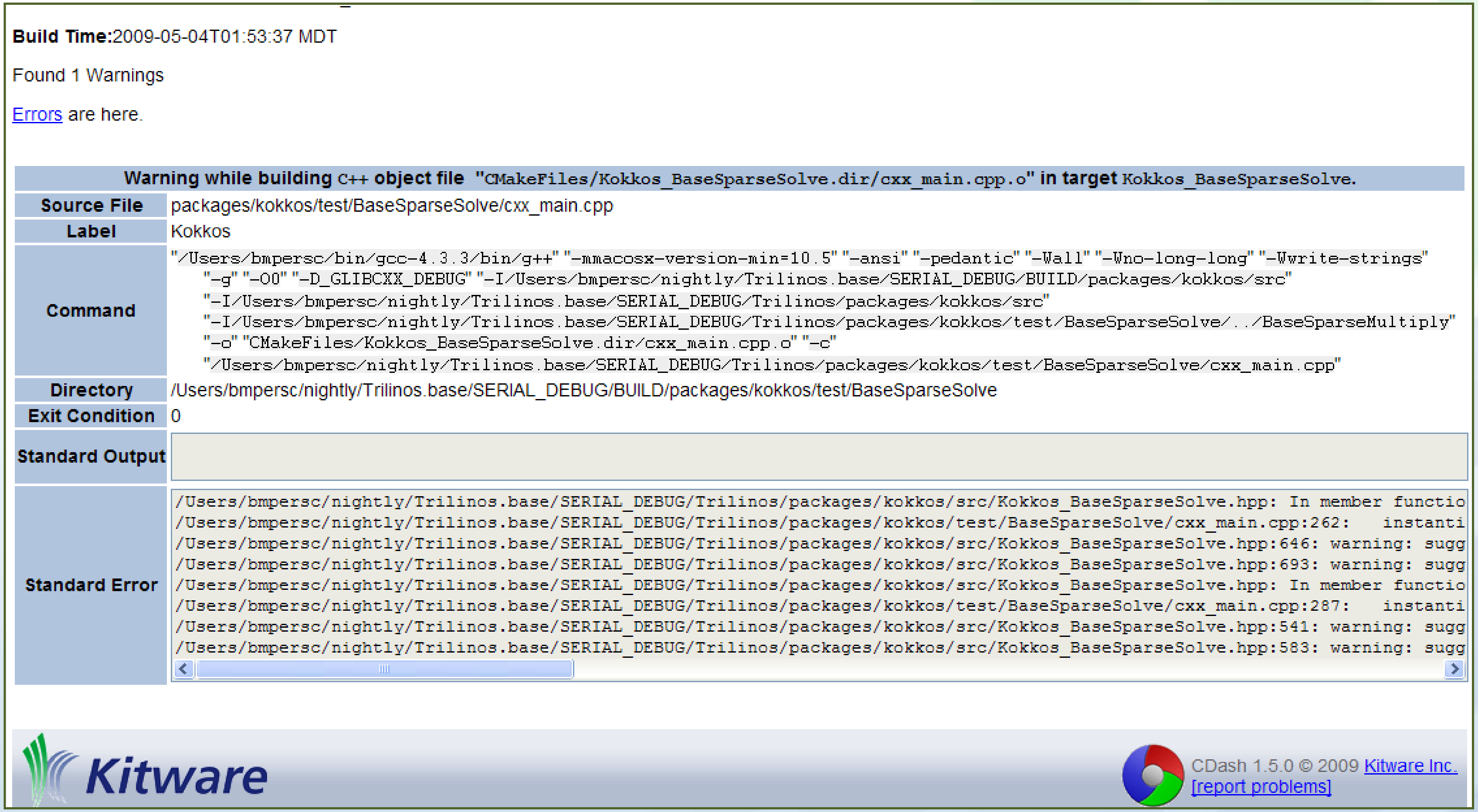
Details on the submission can be found at http://www.cdash.org/CDash/buildSummary.php?buildid=322849

Project: CMake
Site: destiny.kitware
Build Name: HP-UX-aCC
Build Time: 2009-04-29T14:28:00 EDT
Type: Continuous
Tests failing: 85

Tests failing (first 5)
SystemInformationNew (http://www.cdash.org/CDash/testDetails.php?test=21959894&build=322849)
CommandLineTest (http://www.cdash.org/CDash/testDetails.php?test=21959897&build=322849)
FindPackageTest (http://www.cdash.org/CDash/testDetails.php?test=21959898&build=322849)
FindModulesExecuteAll (http://www.cdash.org/CDash/testDetails.php?test=21959899&build=322849)
StringFileTest (http://www.cdash.org/CDash/testDetails.php?test=21959900&build=322849)

-CDash on www.cdash.org

CTest Command Wrappers Output



Build Time:2009-05-04T01:53:37 MDT

Found 1 Warnings

Errors are here.

Warning while building `C++` **object file** "`CMakeFiles/Kokkos_BaseSparseSolve.dir/cxx_main.cpp.o`" **in target** `Kokkos_BaseSparseSolve`.

Source File packages/kokkos/test/BaseSparseSolve/cxx_main.cpp

Label Kokkos

Command

```
"/Users/bmpersc/bin/gcc-4.3.3/bin/g++" "-mmacosx-version-min=10.5" "-ansi" "-pedantic" "-Wall" "-Wno-long-long" "-Wwrite-strings"
    "-g" "-O0" "-D_GLIBCXX_DEBUG" "-I/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/BUILD/packages/kokkos/src"
    "-I/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src"
    "-I/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/test/BaseSparseSolve/../BaseSparseMultiply"
    "-o" "CMakeFiles/Kokkos_BaseSparseSolve.dir/cxx_main.cpp.o" "-c"
    "/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/test/BaseSparseSolve/cxx_main.cpp"
```

Directory /Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/BUILD/packages/kokkos/test/BaseSparseSolve

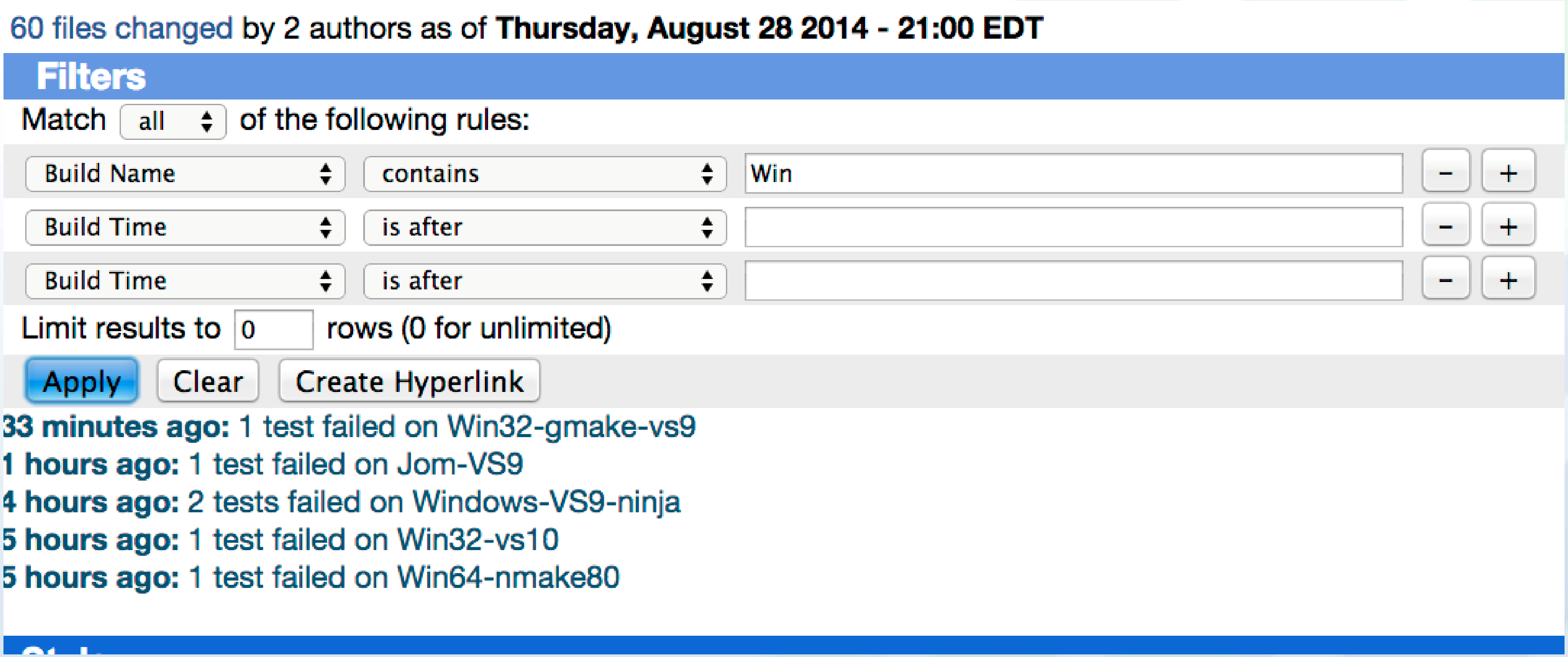
Exit Condition 0

Standard Output

Standard Error

```
/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src/Kokkos_BaseSparseSolve.hpp: In member functio
/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/test/BaseSparseSolve/cxx_main.cpp:262:   instanti
/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src/Kokkos_BaseSparseSolve.hpp:646: warning: sugg
/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src/Kokkos_BaseSparseSolve.hpp:693: warning: sugg
/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src/Kokkos_BaseSparseSolve.hpp: In member functio
/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/test/BaseSparseSolve/cxx_main.cpp:287:   instanti
/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src/Kokkos_BaseSparseSolve.hpp:541: warning: sugg
/Users/bmpersc/nightly/Trilinos.base/SERIAL_DEBUG/Trilinos/packages/kokkos/src/Kokkos_BaseSparseSolve.hpp:583: warning: sugg
```

Query Filters : customize views



60 files changed by 2 authors as of **Thursday, August 28 2014 - 21:00 EDT**

Filters

Match all of the following rules:

Build Name | contains | Win

Build Time | is after |

Build Time | is after |

Limit results to 0 rows (0 for unlimited)

33 minutes ago: 1 test failed on Win32-gmake-vs9

1 hours ago: 1 test failed on Jom-VS9

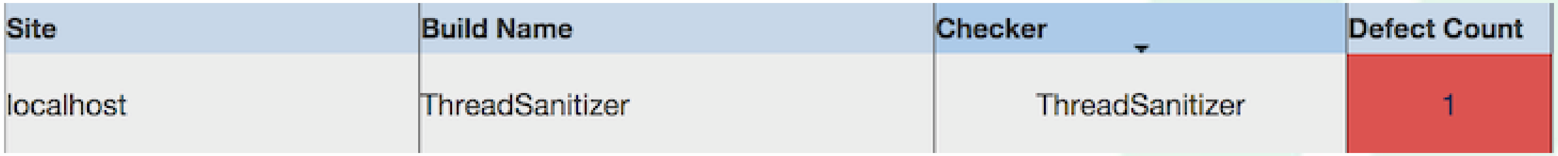
4 hours ago: 2 tests failed on Windows-VS9-ninja

5 hours ago: 1 test failed on Win32-vs10

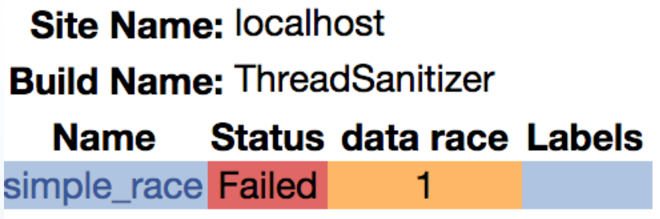
5 hours ago: 1 test failed on Win64-nmake80

Kitware

New clang based tool support



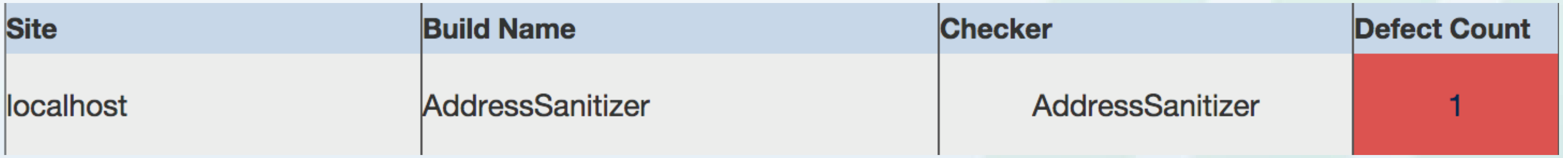
Site	Build Name	Checker	Defect Count
localhost	ThreadSanitizer	ThreadSanitizer	1



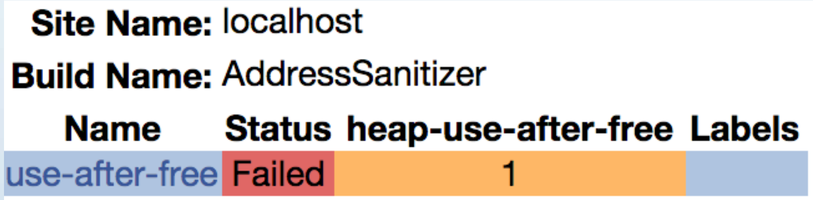
Site Name: localhost

Build Name: ThreadSanitizer

Name	Status	data race	Labels
simple_race	Failed	1	



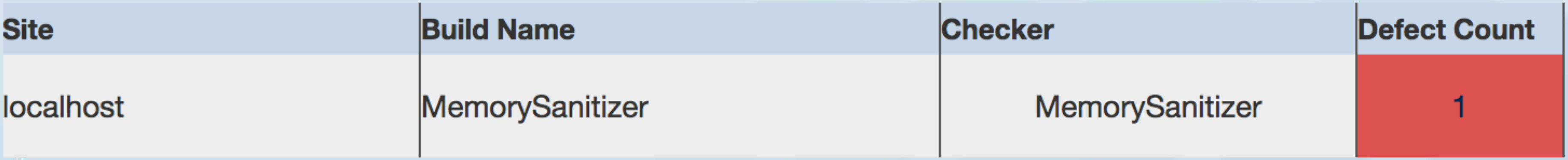
Site	Build Name	Checker	Defect Count
localhost	AddressSanitizer	AddressSanitizer	1



Site Name: localhost

Build Name: AddressSanitizer

Name	Status	heap-use-after-free	Labels
use-after-free	Failed	1	

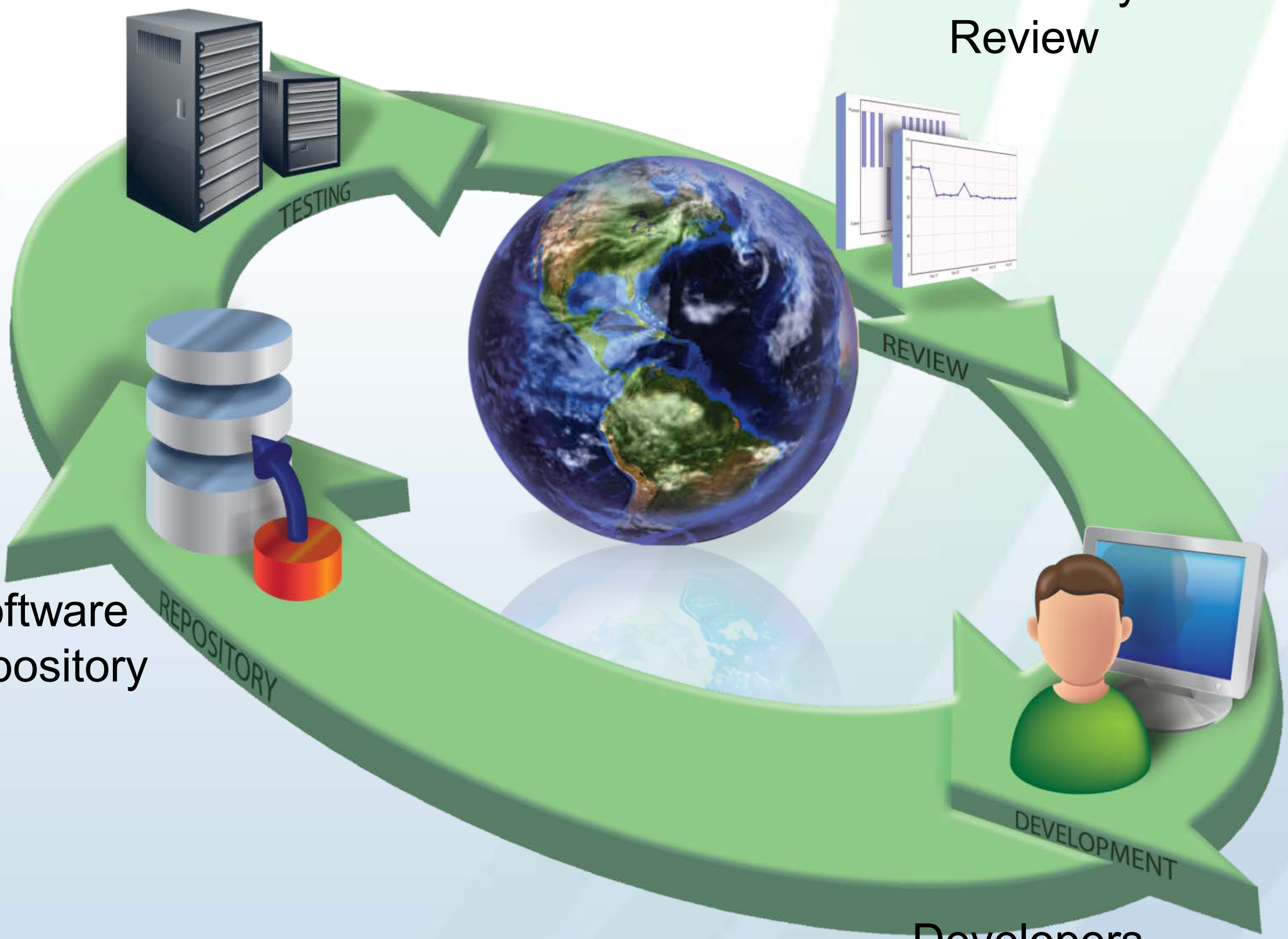


Site	Build Name	Checker	Defect Count
localhost	MemorySanitizer	MemorySanitizer	1

Clang tools cont

- iwyu (include what you use)
- Scanbuild

Software Process – Reproducible Results



Build, Test & Package

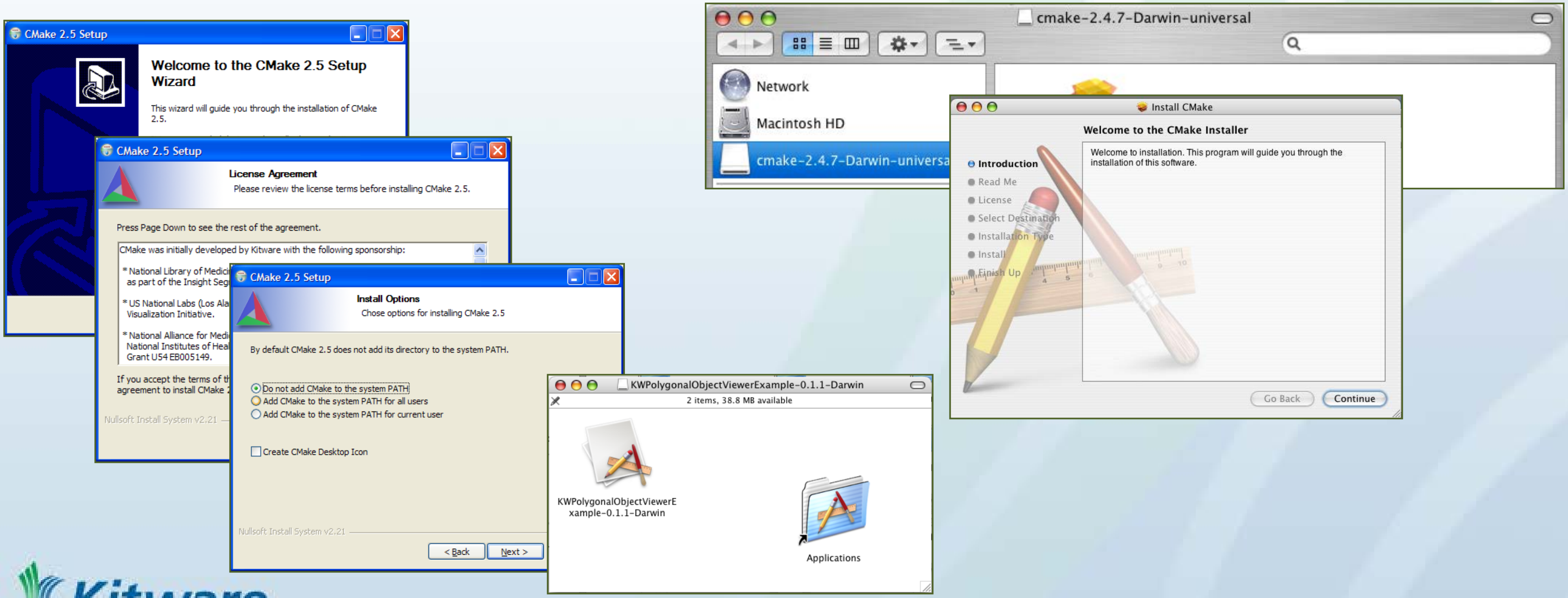
Community Review

Developers & Users

Software Repository

What is CPack

- CPack is bundled with CMake
- Creates professional platform specific installers



This wizard will guide you through the installation of CMake 2.5.

Please review the license terms before installing CMake 2.5.

Press Page Down to see the rest of the agreement.

CMake was initially developed by Kitware with the following sponsorship:

* National Library of Medici
as part of the Insight Seg

* US National Labs (Los Ala
Visualization Initiative.

* National Alliance for Medi
National Institutes of Heal
Grant U54 EB005149.

If you accept the terms of th
agreement to install CMake 2

Chose options for installing CMake 2.5

By default CMake 2.5 does not add its directory to the system PATH.

● Do not add CMake to the system PATH
○ Add CMake to the system PATH for all users
○ Add CMake to the system PATH for current user

□ Create CMake Desktop Icon

Nullsoft Install System v2.21
< Back
Next >
cmake-2.4.7-Darwin-universal
Network
Macintosh HD
cmake-2.4.7-Darwin-universa
KWPolygonalObjectViewerExample-0.1.1-Darwin

2 items, 38.8 MB available

KWPolygonalObjectViewerExample-0.1.1-Darwin
Applications
Install CMake
Welcome to the CMake Installer

Welcome to installation. This program will guide you through the installation of this software.

CPack Features

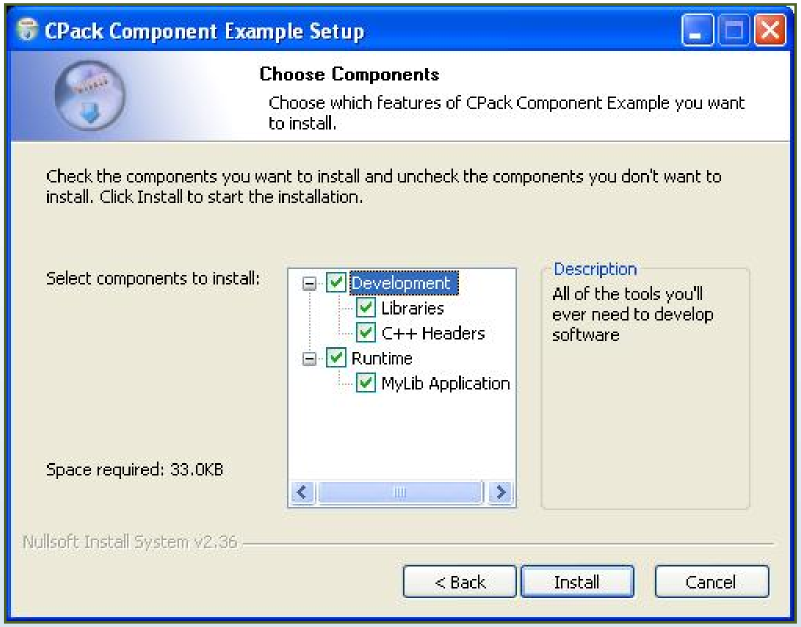
- Supports CMake-based and non-CMake-based projects
- Unix
  - TGZ and self-extracting TGZ (STGZ)
- Windows
  - NullSoft Scriptable Install System (NSIS / NSIS64)
  - WiX
- Mac OSX
  - DragNDrop
  - PackageMaker
- Deb
  - Debian packages
- RPM
  - RPM package manager

Using CPack

- On Windows install command line ZIP program, NSIS and WiX
- Setup your project to work with cpack
  - Get make install to work
    - install(   )
    - make sure you executables work with relative paths and can work from any directory
  - Set cpack option variables if needed
  - include(CPack)

CPack Components

http://www.cmake.org/Wiki/CMake:Component_Install_With_CPack



CPack Component Example Setup

Choose Components

Choose which features of CPack Component Example you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.

Select components to install:

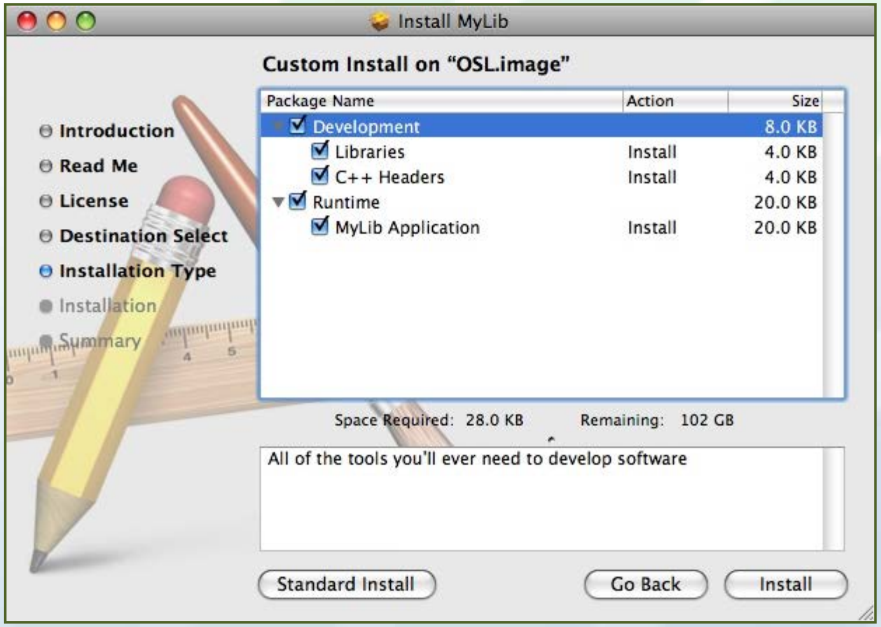
- ☑ Development
  - ☑ Libraries
  - ☑ C++ Headers
- ☑ Runtime
  - ☑ MyLib Application

Description

All of the tools you'll ever need to develop software

Space required: 33.0KB

Nullsoft Install System v2.36



Install MyLib

Custom Install on "OSL.image"

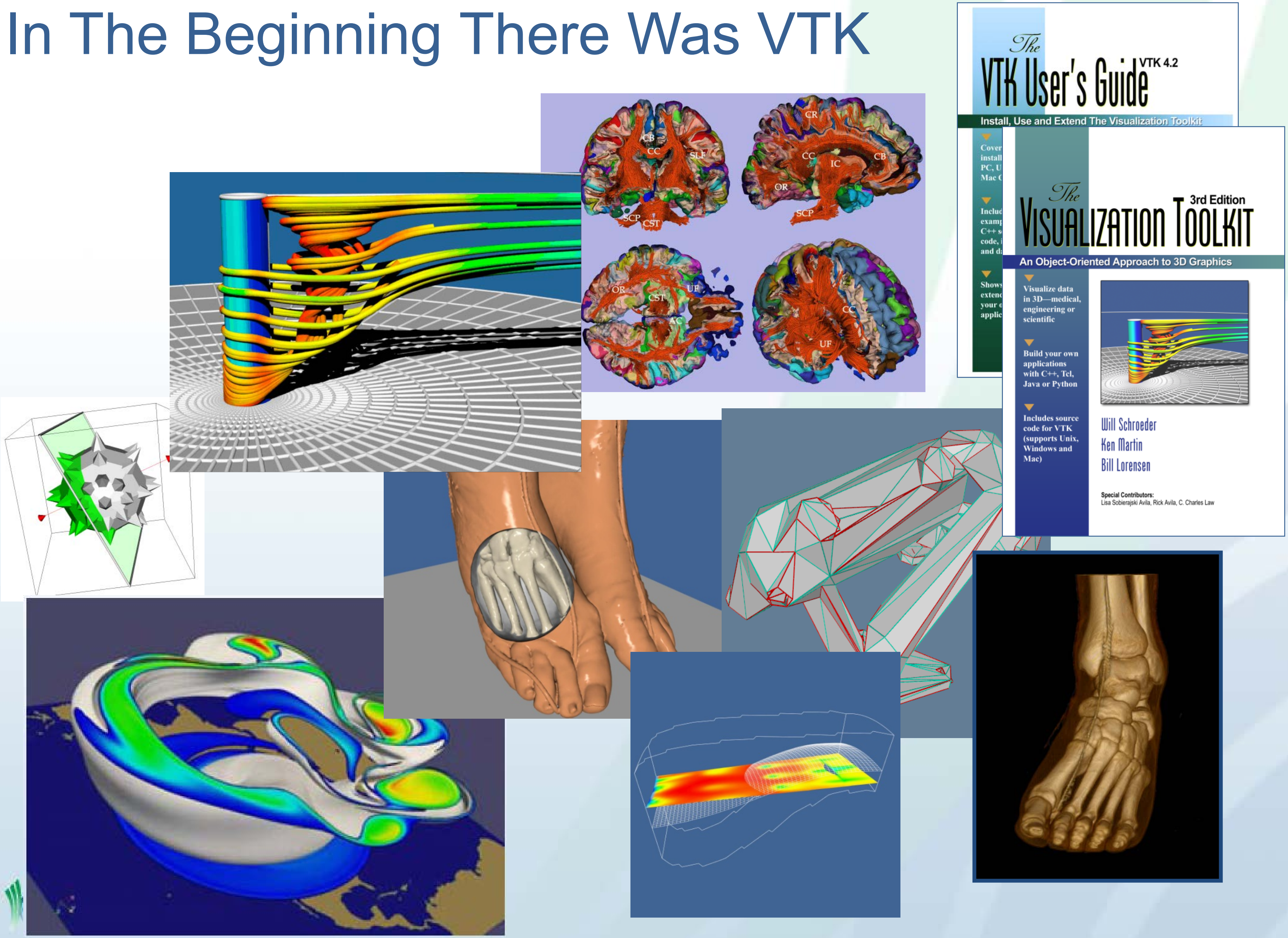
Package Name	Action	Size
☑ Development		8.0 KB
☑ Libraries	Install	4.0 KB
☑ C++ Headers	Install	4.0 KB
☑ Runtime		20.0 KB
☑ MyLib Application	Install	20.0 KB

Space Required: 28.0 KB Remaining: 102 GB

All of the tools you'll ever need to develop software

Examples of CMake Building Science

In The Beginning There Was VTK



The VTK User's Guide VTK 4.2

Install, Use and Extend The Visualization Toolkit

- Cover install PC, U Mac (
- Includ examp C++ s code, i and da
- Shows extend your o applic

The Visualization Toolkit 3rd Edition

An Object-Oriented Approach to 3D Graphics

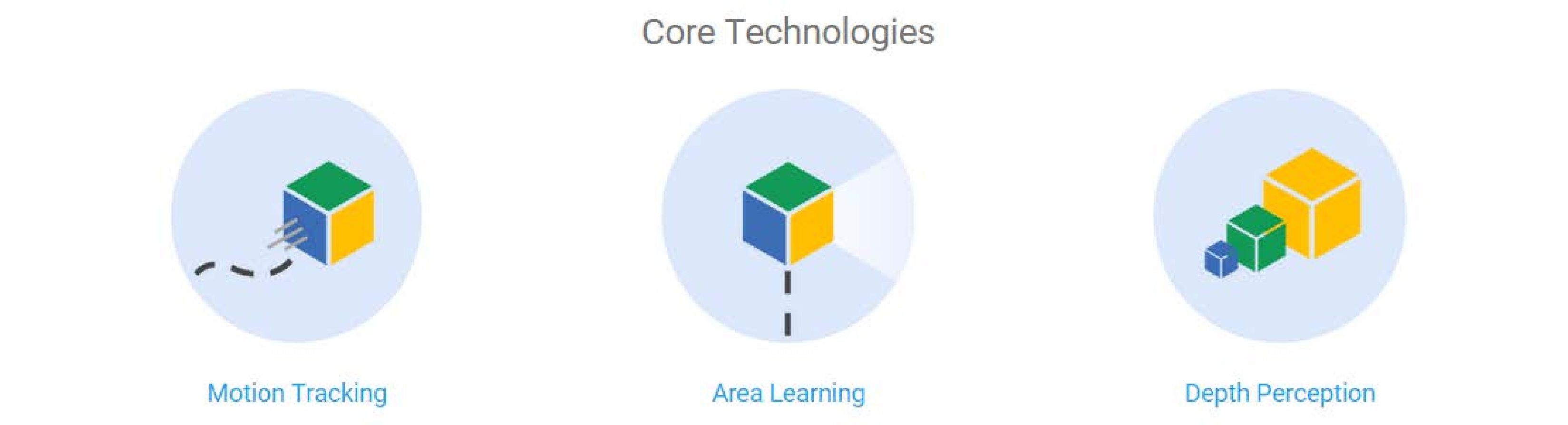
- Visualize data in 3D—medical, engineering or scientific
- Build your own applications with C++, Tcl, Java or Python
- Includes source code for VTK (supports Unix, Windows and Mac)

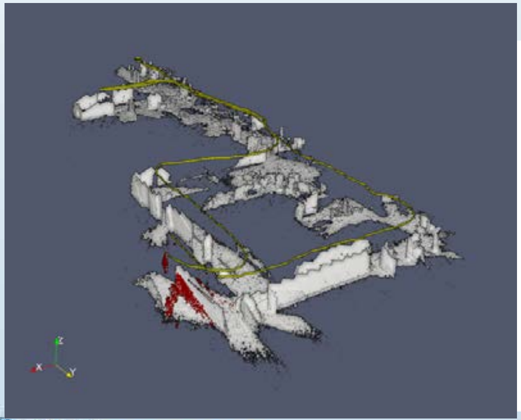
Will Schroeder
Ken Martin
Bill Lorensen

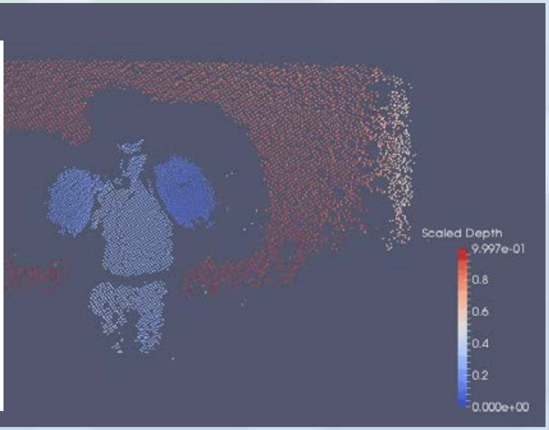
Special Contributors:
Lisa Sobierajski Avila, Rick Avila, C. Charles Law

Google Project Tango

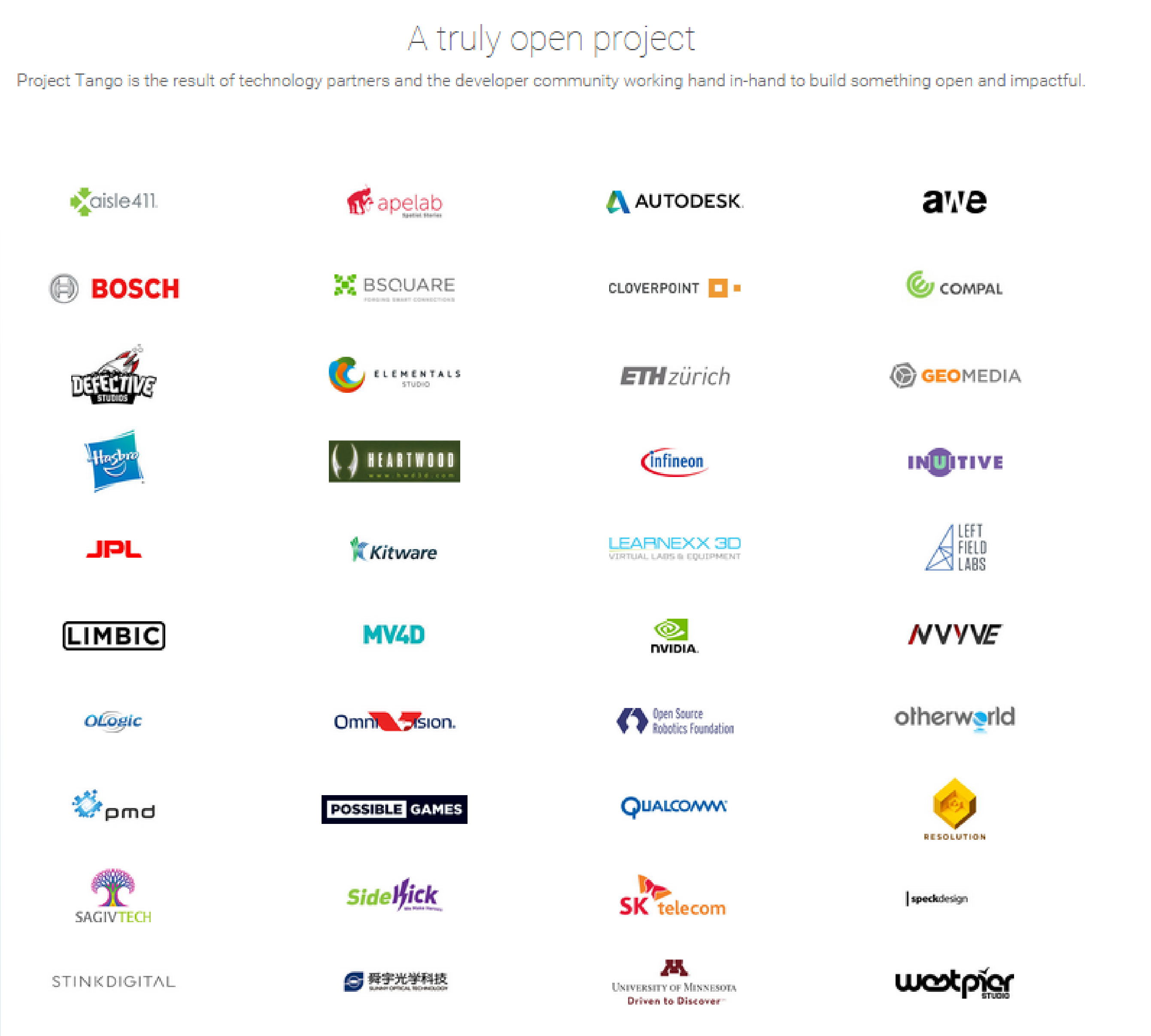
Project Tango technology gives a mobile device the ability to navigate the physical world similar to how we do as humans. Project Tango brings a new kind of spatial perception to the Android device platform by adding advanced computer vision, image processing, and special vision sensors.







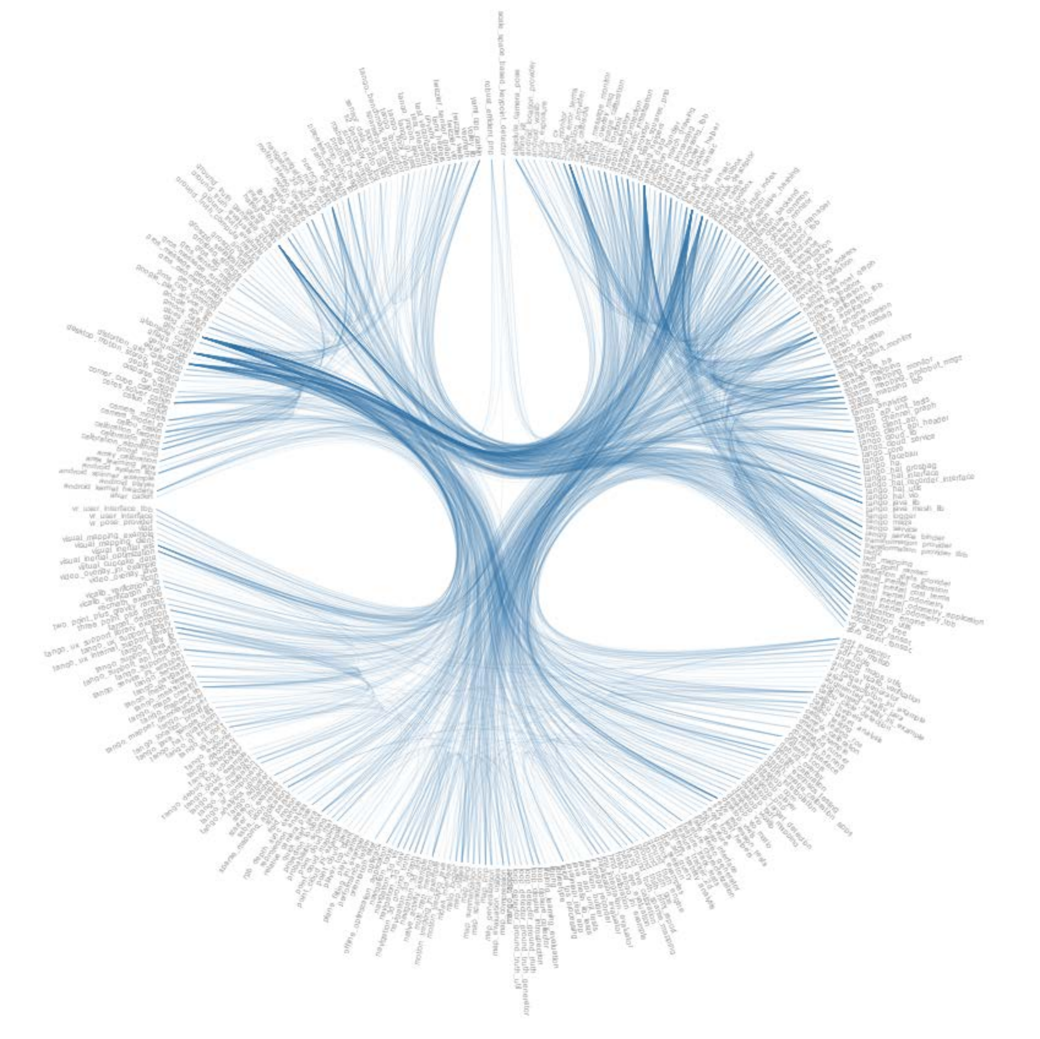
Project Tango: community effort

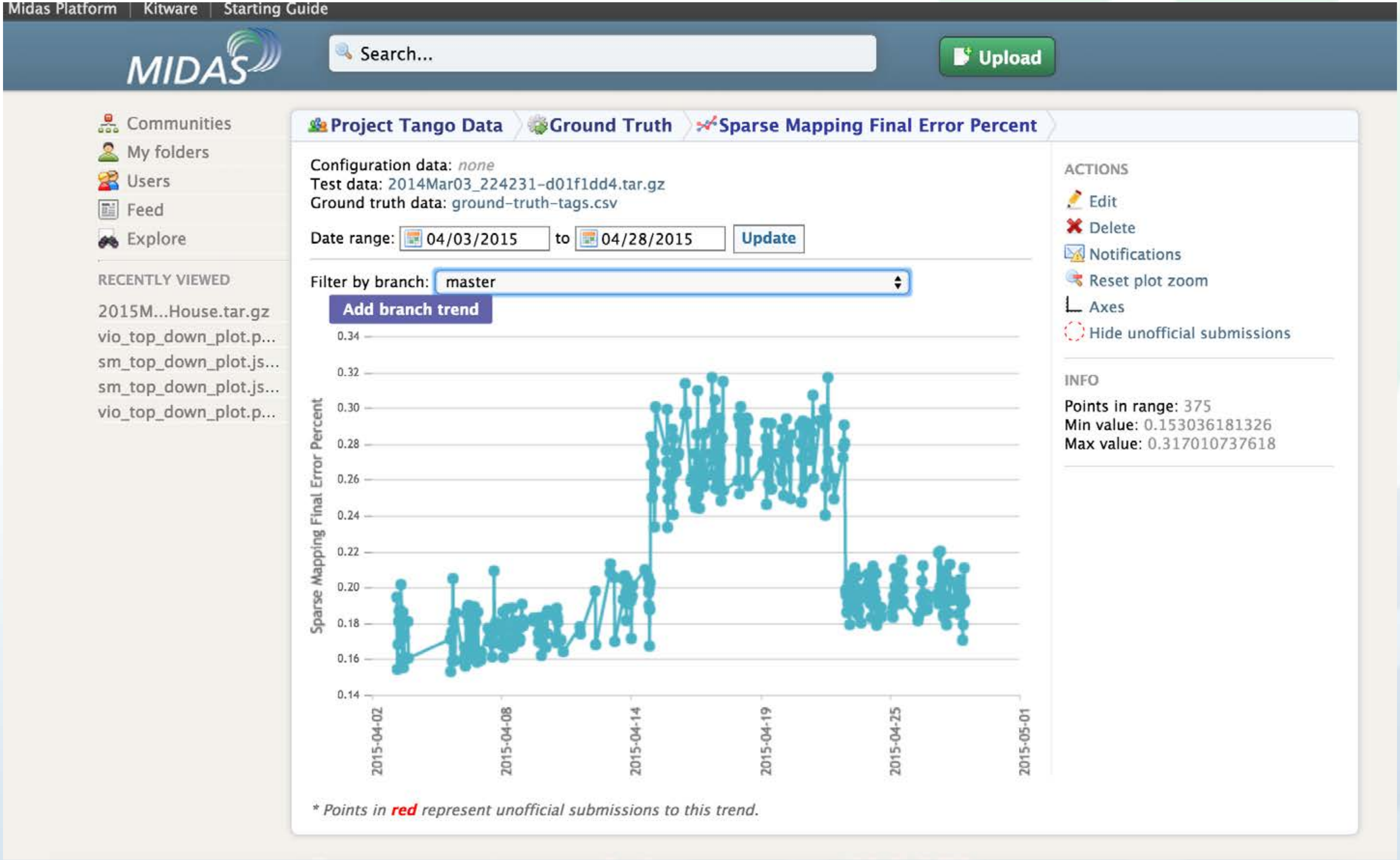


A truly open project

Project Tango is the result of technology partners and the developer community working hand in-hand to build something open and impactful.

SubProject Dependencies Graph for Google Tango from CDash





Configuration data: *none*
Test data: 2014Mar03_224231-d01f1dd4.tar.gz
Ground truth data: ground-truth-tags.csv

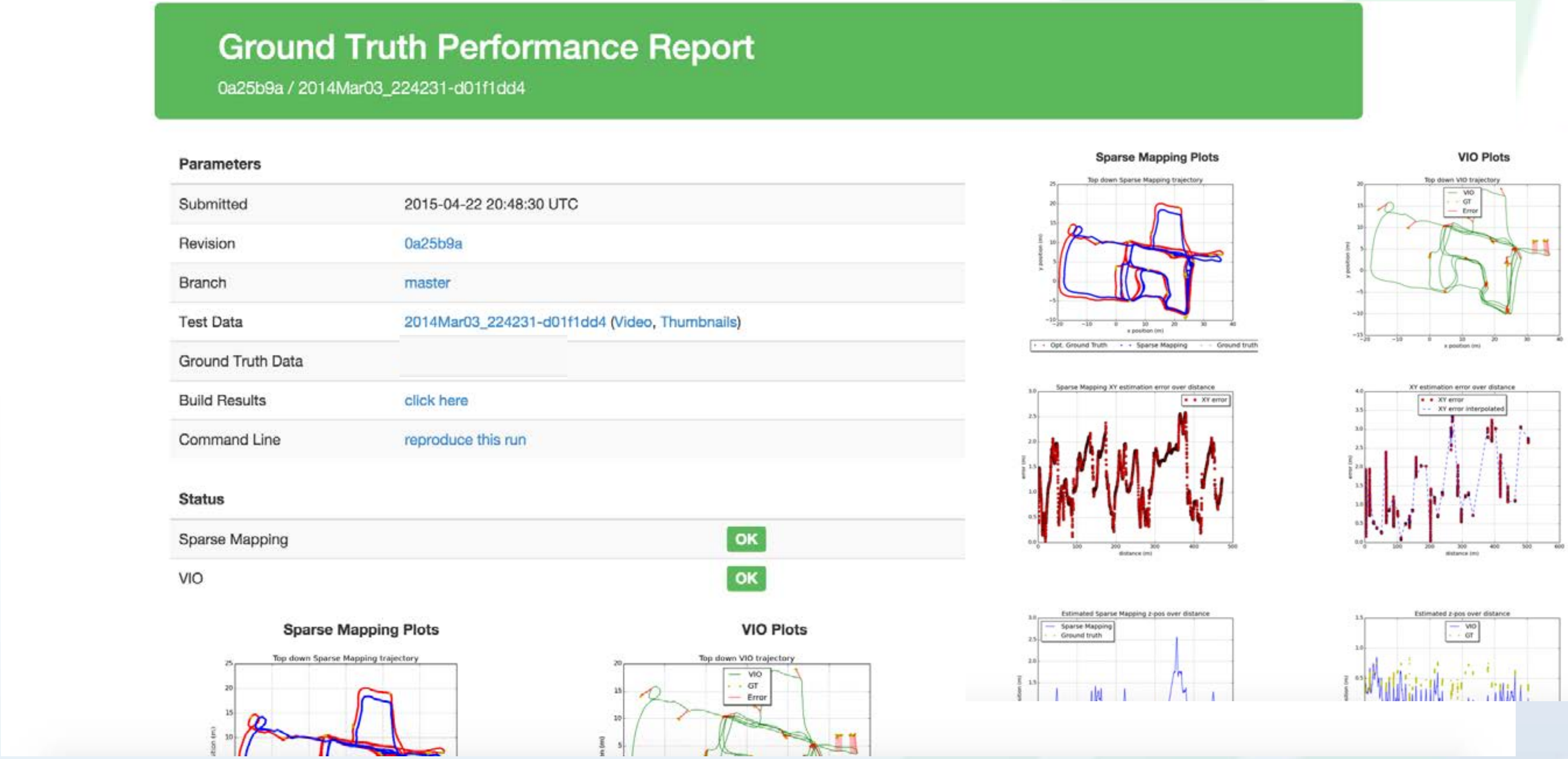
Date range: 04/03/2015 to 04/28/2015

Filter by branch: master

Points in range: 375
Min value: 0.153036181326
Max value: 0.317010737618

** Points in **red** represent unofficial submissions to this trend.*

Kitware



Ground Truth Performance Report

0a25b9a / 2014Mar03_224231-d01f1dd4

Parameters

Submitted	2015-04-22 20:48:30 UTC
Revision	0a25b9a
Branch	master
Test Data	2014Mar03_224231-d01f1dd4 (Video, Thumbnails)
Ground Truth Data	
Build Results	click here
Command Line	reproduce this run

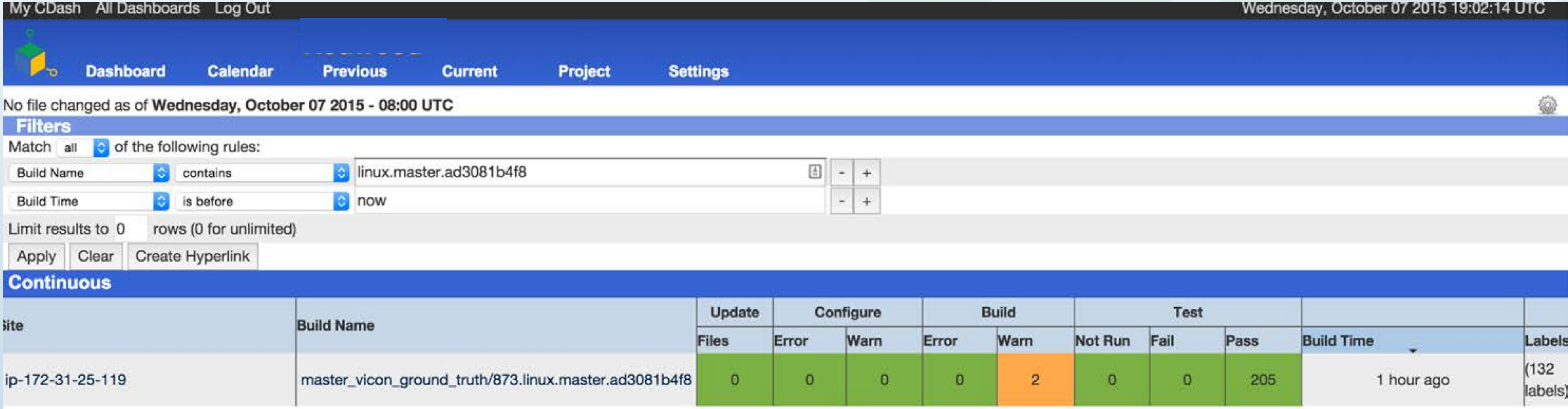
Status

Sparse Mapping	OK
VIO	OK

Sparse Mapping Plots

VIO Plots

Top down Sparse Mapping trajectory
Opt. Ground Truth
Sparse Mapping
Ground truth
Top down VIO trajectory
VIO
GT
Error
Sparse Mapping XY estimation error over distance
XY error
XY estimation error over distance
XY error
XY error interpolated
Estimated Sparse Mapping z-pos over distance
Sparse Mapping
Ground truth
Estimated z-pos over distance
VIO
GT



Wednesday, October 07 2015 19:02:14 UTC

No file changed as of **Wednesday, October 07 2015 - 08:00 UTC**

Filters

Match all of the following rules:

Build Name contains linux.master.ad3081b4f8

Build Time is before now

Limit results to 0 rows (0 for unlimited)

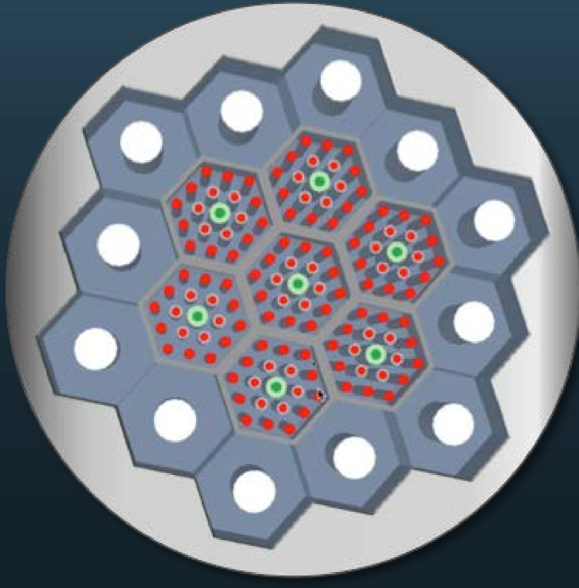
Continuous

ite	Build Name	Update	Configure		Build		Test				Label
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	Build Time	Labels
ip-172-31-25-119	master_vicon_ground_truth/873.linux.master.ad3081b4f8	0	0	0	0	2	0	0	205	1 hour ago	(132 labels)

Kitware

Kitware

Computational Model Builder and The Reactor Geometry Generator Projects

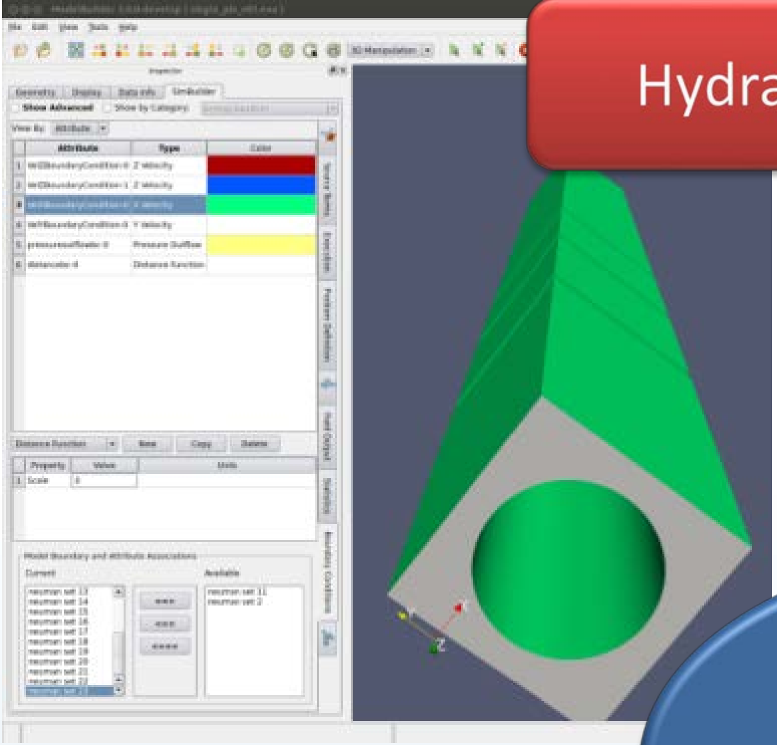


Computational Model Builder

- Goals:
  - Develop an application framework that can be easily adapted to specific problem domains
    - Geometric Domain and Assembly Model Creation
    - Simulation Creation
    - Simulation Execution
  - Quickly create custom applications/workflows
  - Leverage existing and future simulators, meshers, and toolkits
  - Develop a simulation model toolkit (SMTK) analogous to VTK for supporting simulation models
  - All development under BSD licensing

Simulation Codes Currently Supported

Hydra



Inspector

Geometry | Display | Data Info | SimBuilder

□ Show Advanced □ Show by Category

View By: Attribute

	Attribute	Type	Color
1	WallBoundaryCondition-0	Z Velocity	🟥
2	WallBoundaryCondition-1	Z Velocity	🟦
3	WallBoundaryCondition-0	X Velocity	🟩
4	WallBoundaryCondition-0	Y Velocity	
5	pressureoutflowbc-0	Pressure Outflow	🟨
6	distancebc-0	Distance Function	

Distance Function | New | Copy | Delete

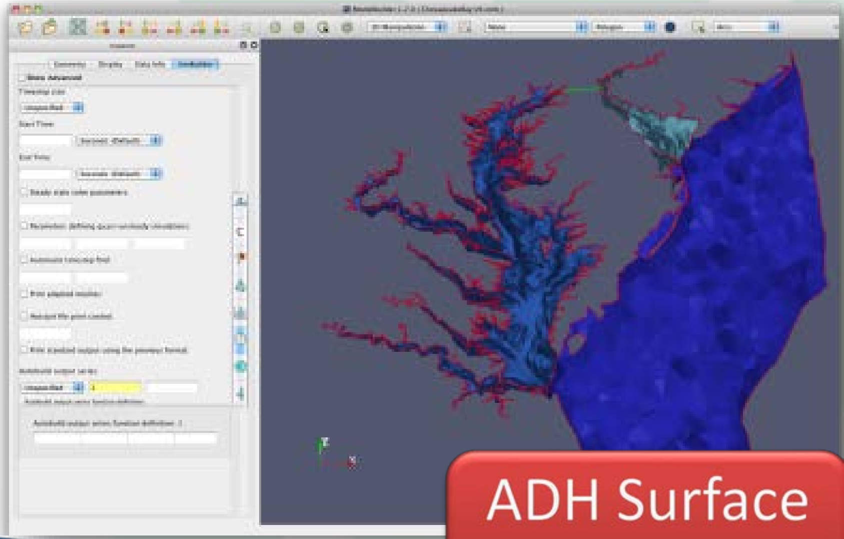
	Property	Value	Units
1	Scale	0	

Model Boundary and Attribute Associations

Current: neumann set 13, neumann set 14, neumann set 15, neumann set 16, neumann set 17, neumann set 18, neumann set 19, neumann set 20, neumann set 21, neumann set 22, neumann set 23

Available: neumann set 11, neumann set 2

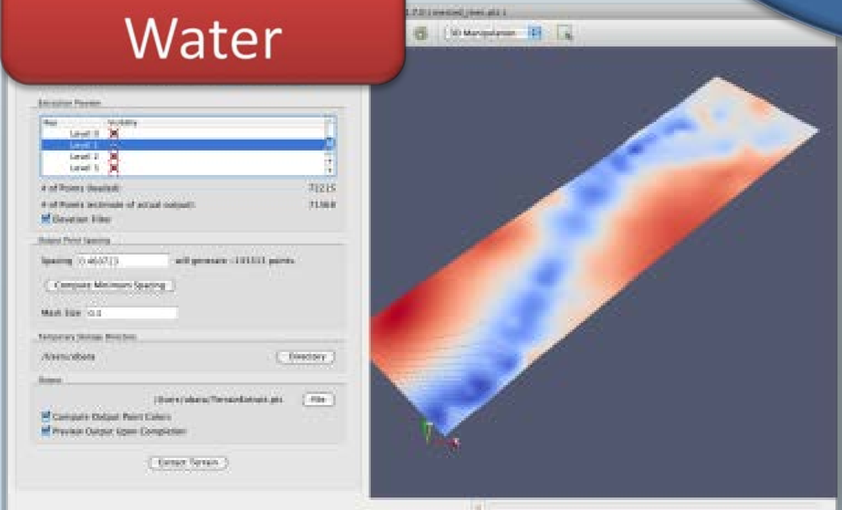
ADH Surface Water



Geometry | Display | Data Info | SimBuilder

□ Show Advanced

ADH Ground Water



Elevation Pyramid

Map	Visibility
Level 0	☒
Level 1	
Level 2	☒
Level 3	☒

# of Points (Seeded) 71215

# of Points (estimate of actual output) 71568

☑ Elevation Filter

Output Point Spacing

Spacing [illegible] will generate ~131515 points

Compute Minimum Spacing

Mesh Size 0.0

Temporary Storage Directory

/Users/albans | Directory

[illegible]

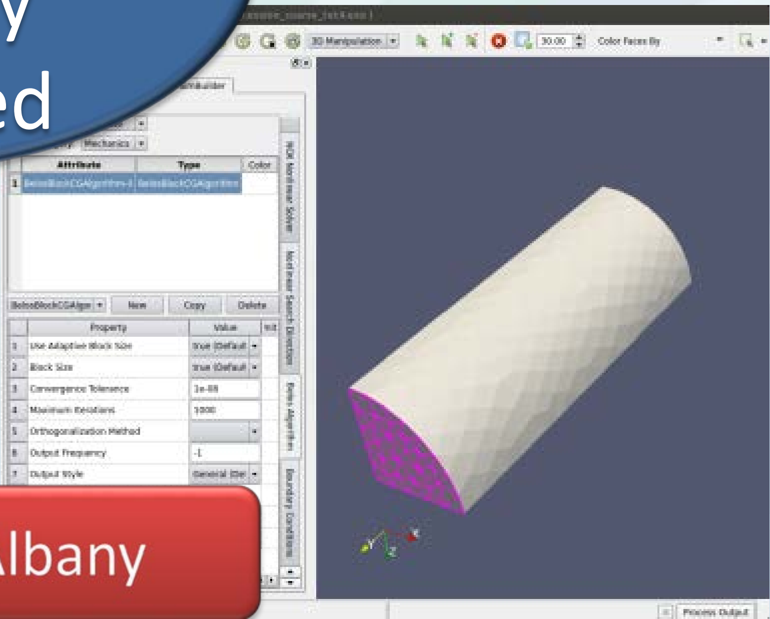
/Users/albans/TerrainExtract.pts | File

☑ Compute Output Point Colors

☑ Preview Output Upon Completion

Extract Terrain

Albany



Mechanics

	Attribute	Type	Color
1	BelosBlockCGAlgorithm-0	BelosBlockCGAlgorithm	

BelosBlockCGAlgo | New | Copy | Delete

	Property	Value	Unit
1	Use Adaptive Block Size	true (Default)	
2	Block Size	true (Default)	
3	Convergence Tolerance	1e-08	
4	Maximum Iterations	1000	
5	Orthogonalization Method		
6	Output Frequency	-1	
7	Output Style	General Ste	

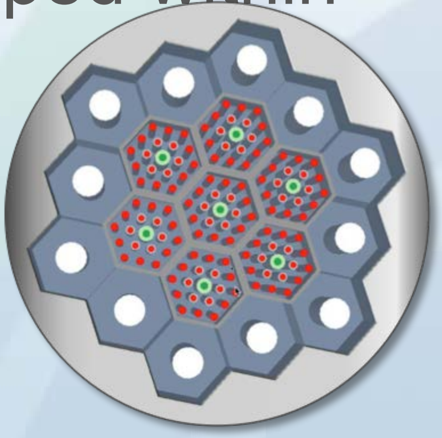
Libraries Used in CMB

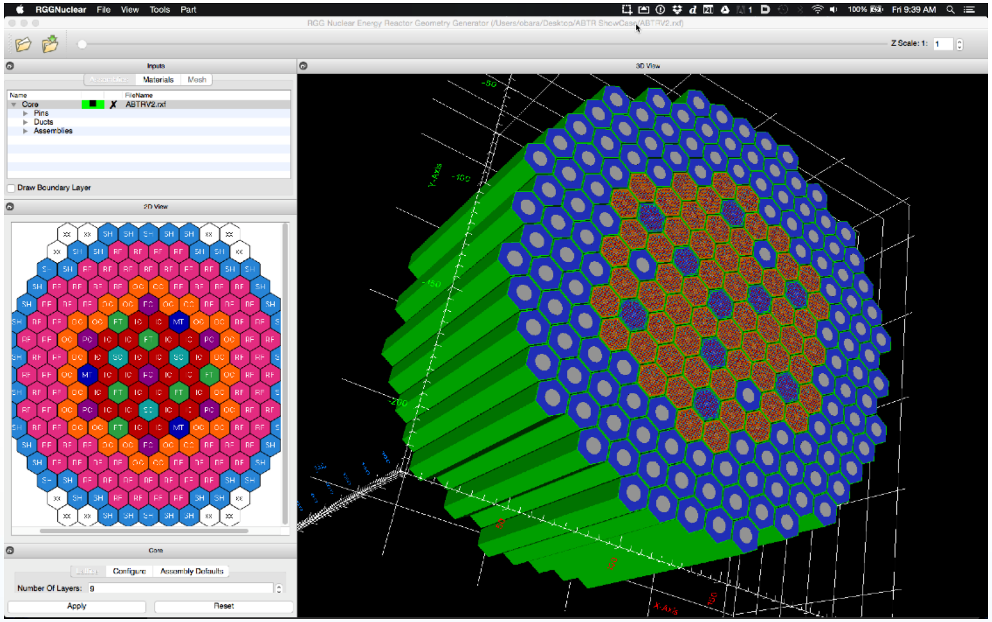
- SMTK
- VTK
- ParaView
- QT
- VXL
- MOAB
- CGM
- OpenCascade
- Boost
- ReMUs
- ZeroMQ
- Shiboken
- Lasso
- HDF5
- NetCDF
- Zlib
- PNG
- Szip
- Pugi XML
- MPI
- Gdal
- Ftgl

Reactor Geometry Generator (RGG)

Goals:

- Provide a set of easy to use model and mesh generation components/applications that support various Nuclear Energy Workflows
- Support Client/Server Framework as well as HPC Environments
- Leverage existing efforts being developed within the DOE




RGGNuclear
File
View
Tools
Part
Fri 9:39 AM
Z Scale: 1:
Inputs
Assemblies
Materials
Mesh
2D View
3D View
Core
Lattice
Configure
Assembly Defaults
Apply
Reset
XX XX SH SH SH SH SH XX XX
XX SH SH RF RF RF RF SH SH XX
SH SH RF RF RF RF RF RF RF SH SH
SH RF RF RF RF OC OC RF RF RF RF SH
SH RF RF RF OC OC PC OC OC RF RF RF SH
SH RF RF OC OC FT IC IC MT OC OC RF RF S
RF RF OC PC IC IC FT IC IC PC OC RF RF
SH RF RF OC IC SC IC IC SC IC OC RF RF S
RF RF OC MT IC IC PC IC IC FT OC RF RF
SH RF RF OC IC FT IC IC FT IC OC RF RF S
RF RF OC PC IC IC SC IC IC PC OC RF RF
SH RF RF OC OC FT IC IC MT OC OC RF RF S
SH RF RF RF OC OC PC OC OC RF RF RF SH
SH RF RF RF RF OC OC RF RF RF RF SH
SH SH RF RF RF RF RF RF RF SH SH
XX SH SH RF RF RF RF SH SH XX
XX XX SH SH SH SH SH XX XX

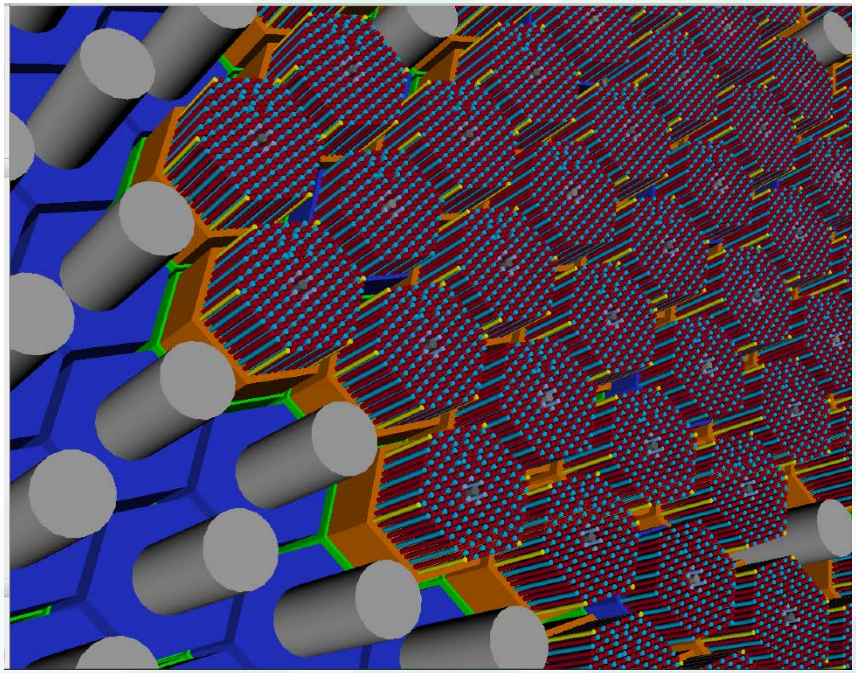
RGG Nuclear Energy Reactor Geometry Generator (/Users/obara/Desktop/ABTR ShowCase/ABTRV2.rxf)

Z Scale: 1: 1

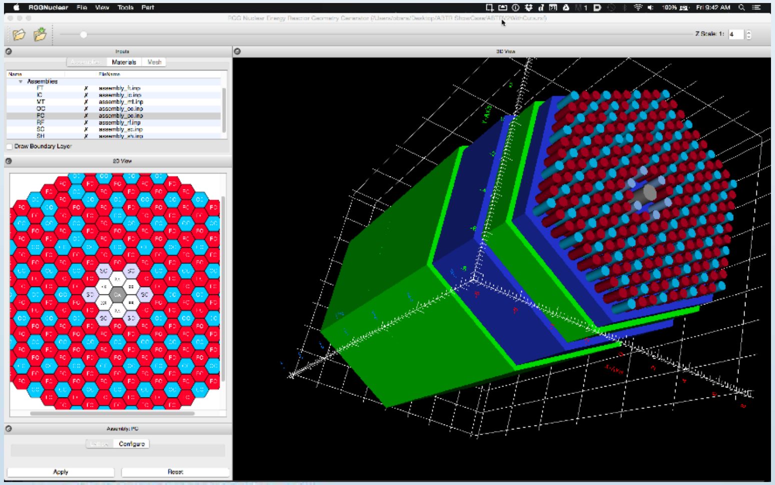
Name			FileName
Core	🟩	✗	ABTRV2.rxf
Pins			
Ducts			
Assemblies			

□ Draw Boundary Layer

Number Of Layers: 9



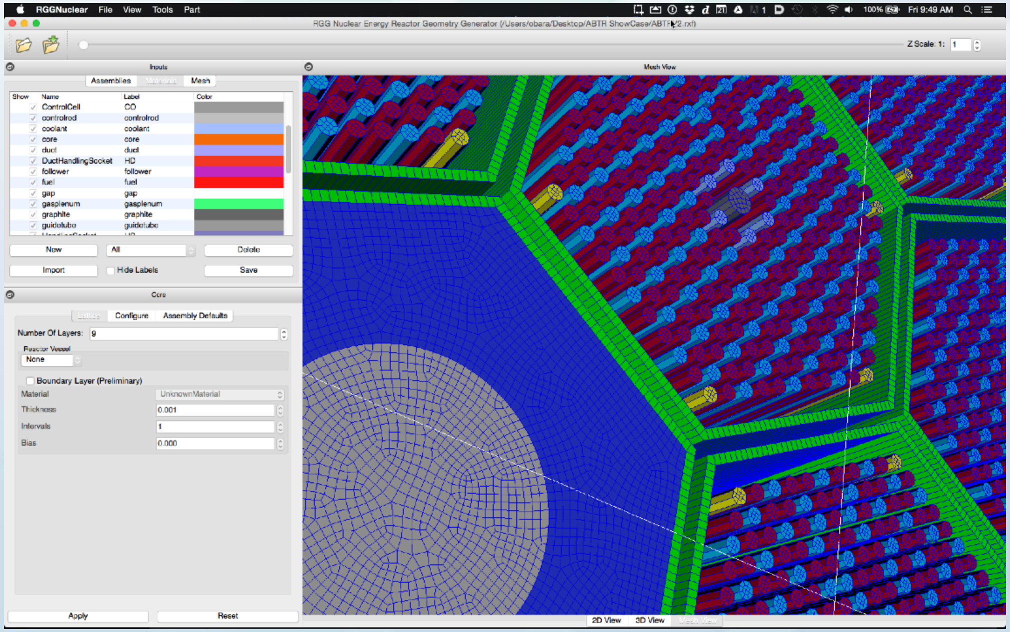
Model of a Sodium Cooled Fast Reactor Core Modeled & Meshed in RGG



Z Scale: 1: 4

Name		FileName
Assemblies		
FT	✗	assembly_ft.inp
IC	✗	assembly_ic.inp
MT	✗	assembly_mt.inp
OC	✗	assembly_oc.inp
PC	✗	assembly_pc.inp
RF	✗	assembly_rf.inp
SC	✗	assembly_sc.inp
SH	✗	assembly_sh.inp

□ Draw Boundary Layer



RGG Nuclear Energy Reactor Geometry Generator (/Users/obara/Desktop/ABTR ShowCase/ABTRV2.rxf)

Z Scale: 1: 1

Show	Name	Label	Color
✓	ControlCell	CO	⬜
✓	controlrod	controlrod	⬜
✓	coolant	coolant	🟦
✓	core	core	🟧
✓	duct	duct	🟪
✓	DuctHandlingSocket	HD	🟥
✓	follower	follower	🟪
✓	fuel	fuel	🟥
✓	gap	gap	
✓	gasplenum	gasplenum	🟩
✓	graphite	graphite	⬛
✓	guidetube	guidetube	⬜

□ Hide Labels

Number Of Layers: 9

Reactor Vessel: None

□ Boundary Layer (Preliminary)

Material: UnknownMaterial

Thickness: 0.001

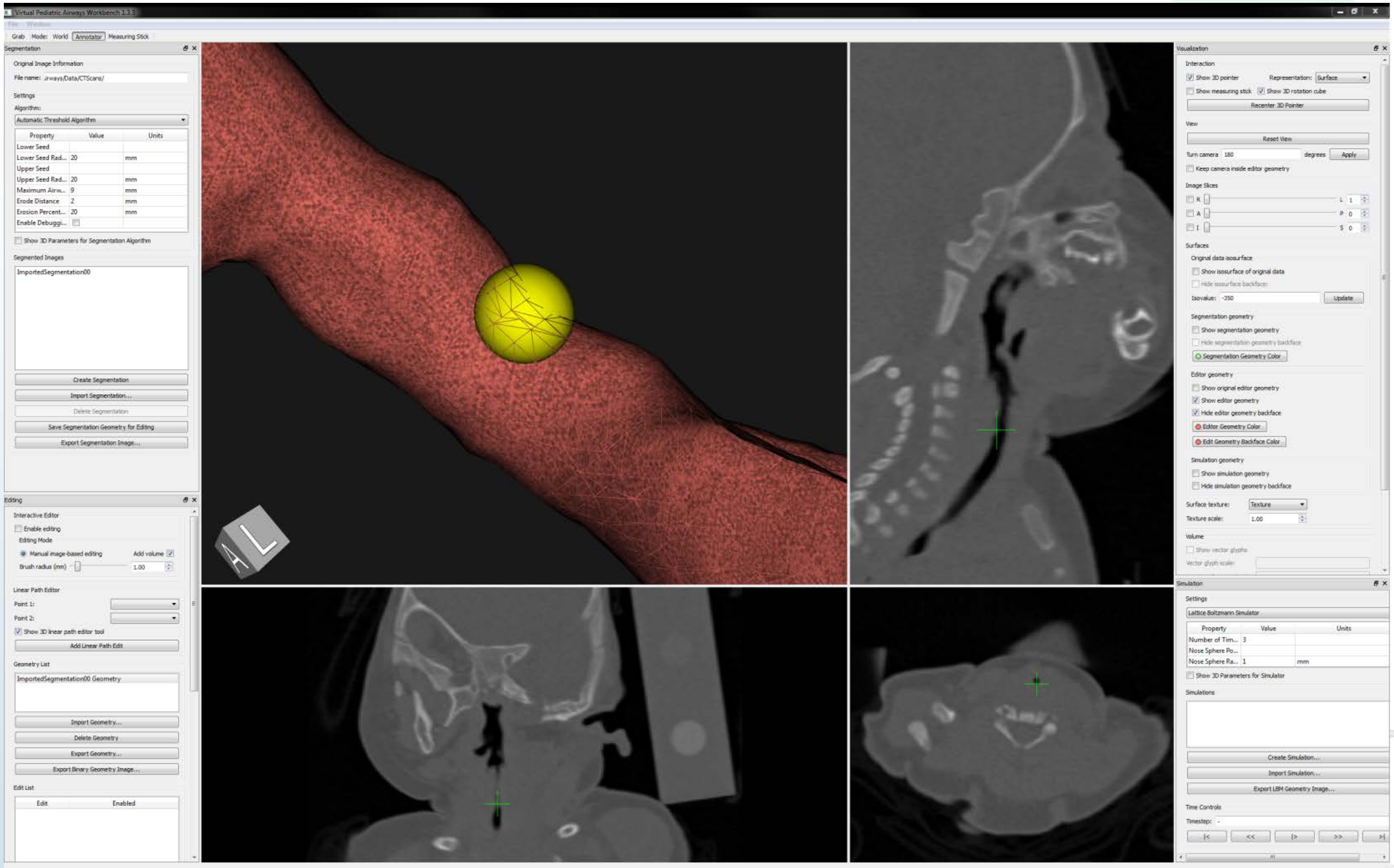
Intervals: 1

Bias: 0.000

Libraries Used in RGG

- VTK
- QT
- MOAB
- CGM
- OpenCascade
- Boost
- ReMUs
- ZeroMQ
- CUBIT
- Lasso
- HDF5
- NetCDF
- Zlib
- PNG
- Szip
- MPI
- Ftgl

Virtual Pediatric Airways Workbench (VPAW)



Original Image Information

File name: .irways/Data/CTScans/

Settings

Algorithm: Automatic Threshold Algorithm

Property	Value	Units
Lower Seed		
Lower Seed Rad...	20	mm
Upper Seed		
Upper Seed Rad...	20	mm
Maximum Airw...	9	mm
Erode Distance	2	mm
Erosion Percent...	20	mm
Enable Debuggi...	□	

□ Show 3D Parameters for Segmentation Algorithm

Segmented Images

ImportedSegmentation00

Interactive Editor

□ Enable editing

Editing Mode

● Manual image-based editing Add volume ☑

Brush radius (mm) 1.00

Linear Path Editor

Point 1:

Point 2:

☑ Show 3D linear path editor tool

Geometry List

ImportedSegmentation00 Geometry

Edit List

Edit	Enabled

Interaction

☑ Show 3D pointer Representation: Surface

□ Show measuring stick ☑ Show 3D rotation cube

View

Turn camera 180 degrees

□ Keep camera inside editor geometry

Image Slices

□ R L 1

□ A P 0

□ I S 0

Surfaces

Original data isosurface

□ Show isosurface of original data

□ Hide isosurface backface

Isovalue: -350

Segmentation geometry

□ Show segmentation geometry

□ Hide segmentation geometry backface

Editor geometry

□ Show original editor geometry

☑ Show editor geometry

☑ Hide editor geometry backface

Simulation geometry

□ Show simulation geometry

□ Hide simulation geometry backface

Surface texture: Texture

Texture scale: 1.00

Volume

□ Show vector glyphs

Vector glyph scale:

Settings

Lattice Boltzmann Simulator

Property	Value	Units
Number of Tim...	3	
Nose Sphere Po...		
Nose Sphere Ra...	1	mm

□ Show 3D Parameters for Simulator

Simulations

Time Controls

Timestep: -

VPAW is a surgical planning system for pediatric upper airway obstructions

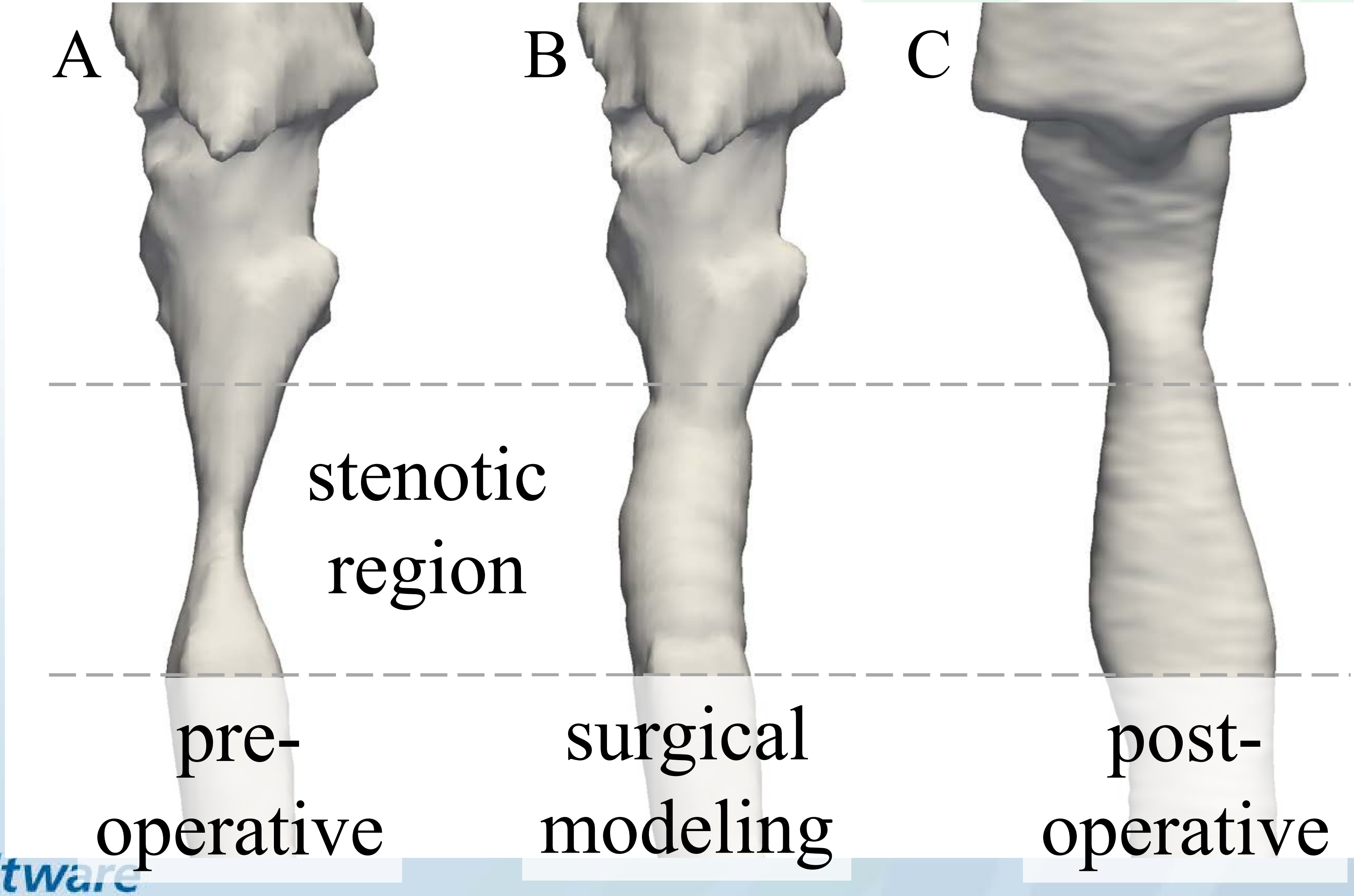
3D interactive editing of airway geometry with 3D input devices to explore possible surgical outcomes

Dependencies built with CMake:

- VTK
- ITK
- VRPN
- quatlib
- jsoncpp
- CLAPACK

Kitware

Case study


A
B
C
stenotic region
pre-operative
surgical modeling
post-operative

ARC Centre of Excellence for Autonomous Systems- Australia

- The Argo is fully automated with the onboard computers all using code built using CMake. The Helicopter is semi-automated with the pilot following a course determined by the onboard computers interacting with the Argo. Everything uses CMake. QNX dashboard

Kitware

Sandia National Laboratories

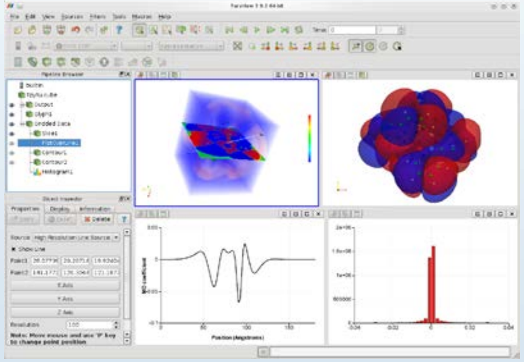
Los Alamos NATIONAL LABORATORY EST.1943

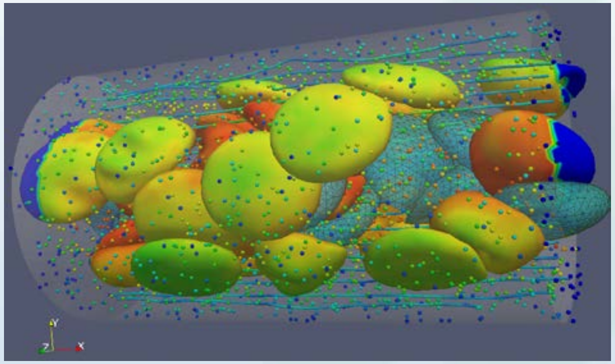
ASC™

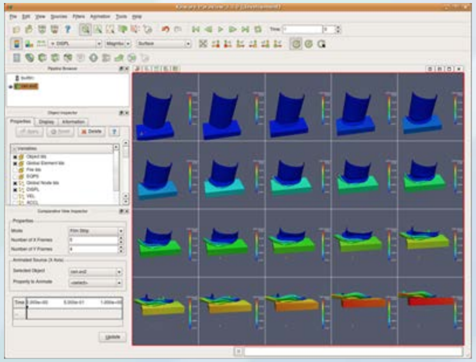
U.S. Army Research Laboratory

ParaView

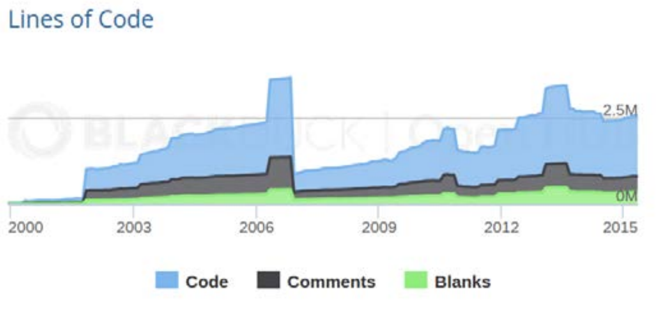
ParaView is an **open-source**, **multi-platform**, **data analysis and visualization** application for analyzing **extremely large datasets** using distributed memory computing resources.



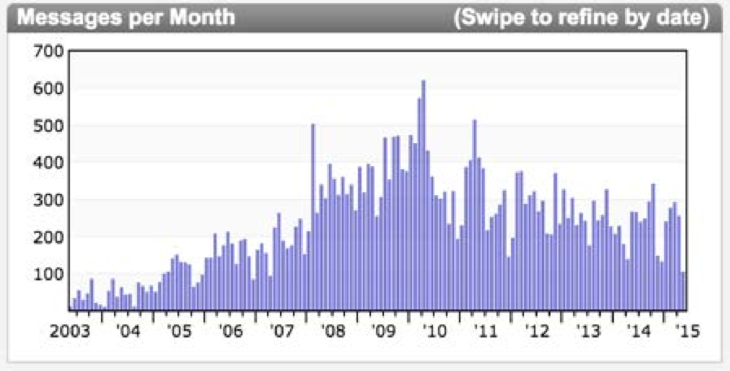


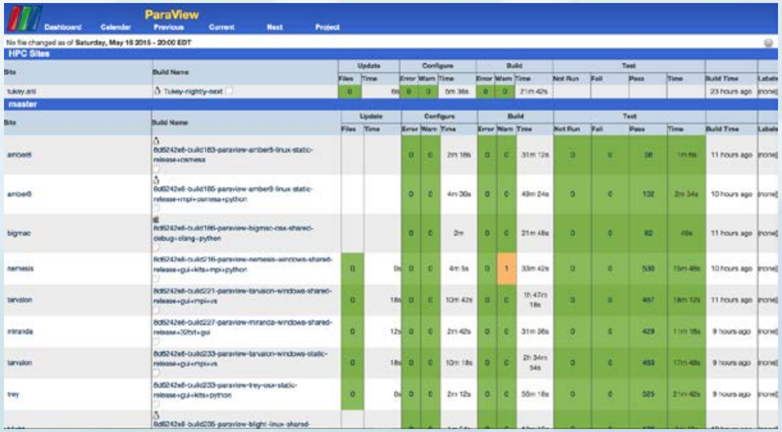


open-source



Grid View





No file changed as of **Saturday, May 16 2015 - 20:00 EDT**

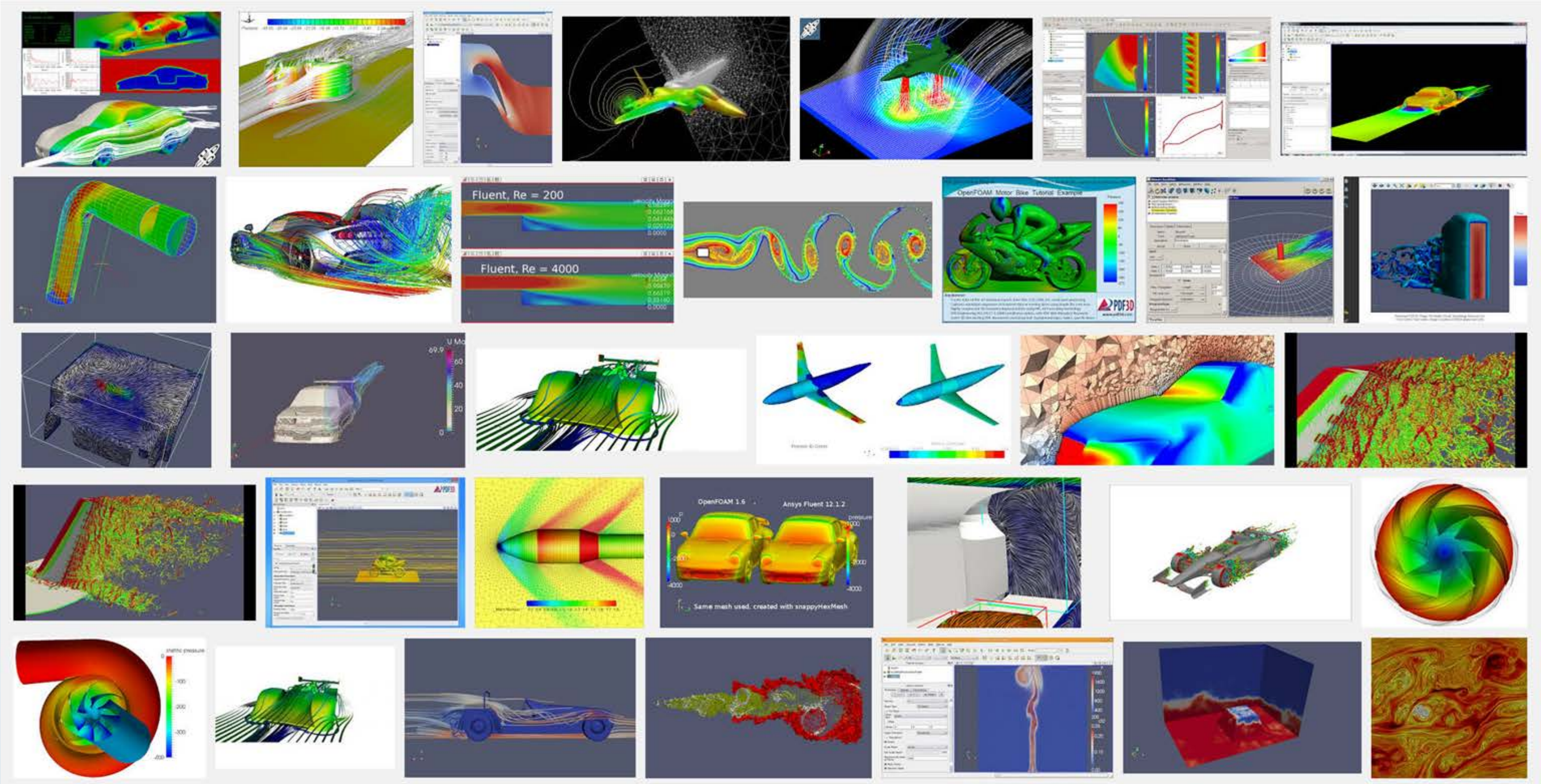
HPC Sites

Site	Build Name	Update		Configure			Build			Test					
		Files	Time	Error	Warn	Time	Error	Warn	Time	Not Run	Fail	Pass	Time	Build Time	Labels
tukey.anl	Tukey-nightly-next	0	6s	0	0	5m 38s	0	0	21m 42s					23 hours ago	(none)

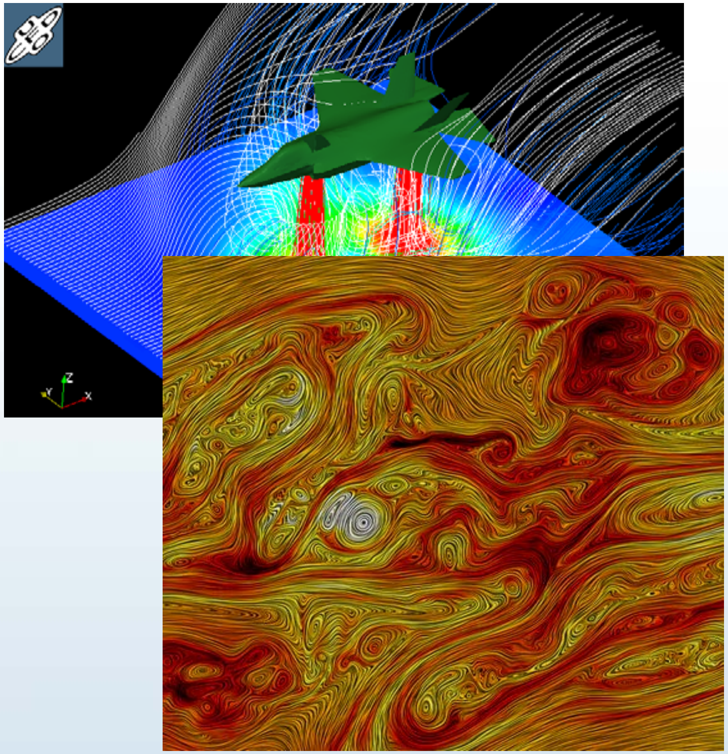
master

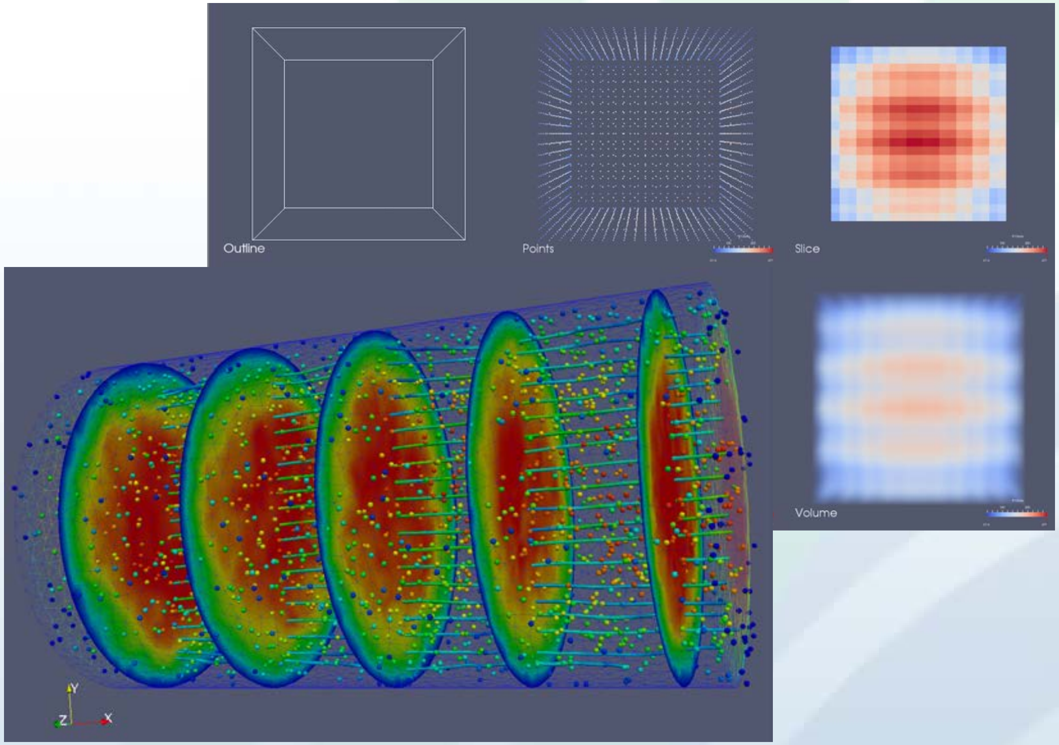
Site	Build Name	Update		Configure			Build			Test					
		Files	Time	Error	Warn	Time	Error	Warn	Time	Not Run	Fail	Pass	Time	Build Time	Labels
amber8	8d6242e8-build183-paraview-amber8-linux-static-release+osmesa			0	0	2m 18s	0	0	31m 12s	0	0	38	1m 6s	11 hours ago	(none)
amber8	8d6242e8-build185-paraview-amber8-linux-static-release+mpi+osmesa+python			0	0	4m 30s	0	0	49m 24s	0	0	132	2m 34s	10 hours ago	(none)
bigmac	8d6242e8-build186-paraview-bigmac-osx-shared-debug+clang+python			0	0	2m	0	0	21m 48s	0	0	82	48s	11 hours ago	(none)
nemesis	8d6242e8-build216-paraview-nemesis-windows-shared-release+gui+kits+mpi+python	0	0s	0	0	4m 5s	0	1	33m 42s	0	0	538	15m 48s	10 hours ago	(none)
tarvalon	8d6242e8-build221-paraview-tarvalon-windows-shared-release+gui+mpi+vs	0	18s	0	0	10m 42s	0	0	1h 47m 18s	0	0	467	18m 12s	11 hours ago	(none)
miranda	8d6242e8-build227-paraview-miranda-windows-shared-release+32bit+gui	0	12s	0	0	2m 42s	0	0	31m 36s	0	0	429	11m 18s	9 hours ago	(none)
tarvalon	8d6242e8-build233-paraview-tarvalon-windows-static-release+gui+mpi+vs	0	18s	0	0	10m 18s	0	0	2h 34m 54s	0	0	453	17m 48s	9 hours ago	(none)
trey	8d6242e8-build233-paraview-trey-osx-static-release+gui+kits+python	0	0s	0	0	2m 12s	0	0	58m 18s	0	0	525	21m 42s	9 hours ago	(none)
	8d6242e8-build235-paraview-blight-linux-shared-														

data analysis & vis


Fluent, Re = 200
Fluent, Re = 4000
OpenFOAM Motor Bike Tutorial Example
PDF3D
OpenFOAM 1.6
Ansys Fluent 12.1.2
Same mesh used, created with snappyHexMesh

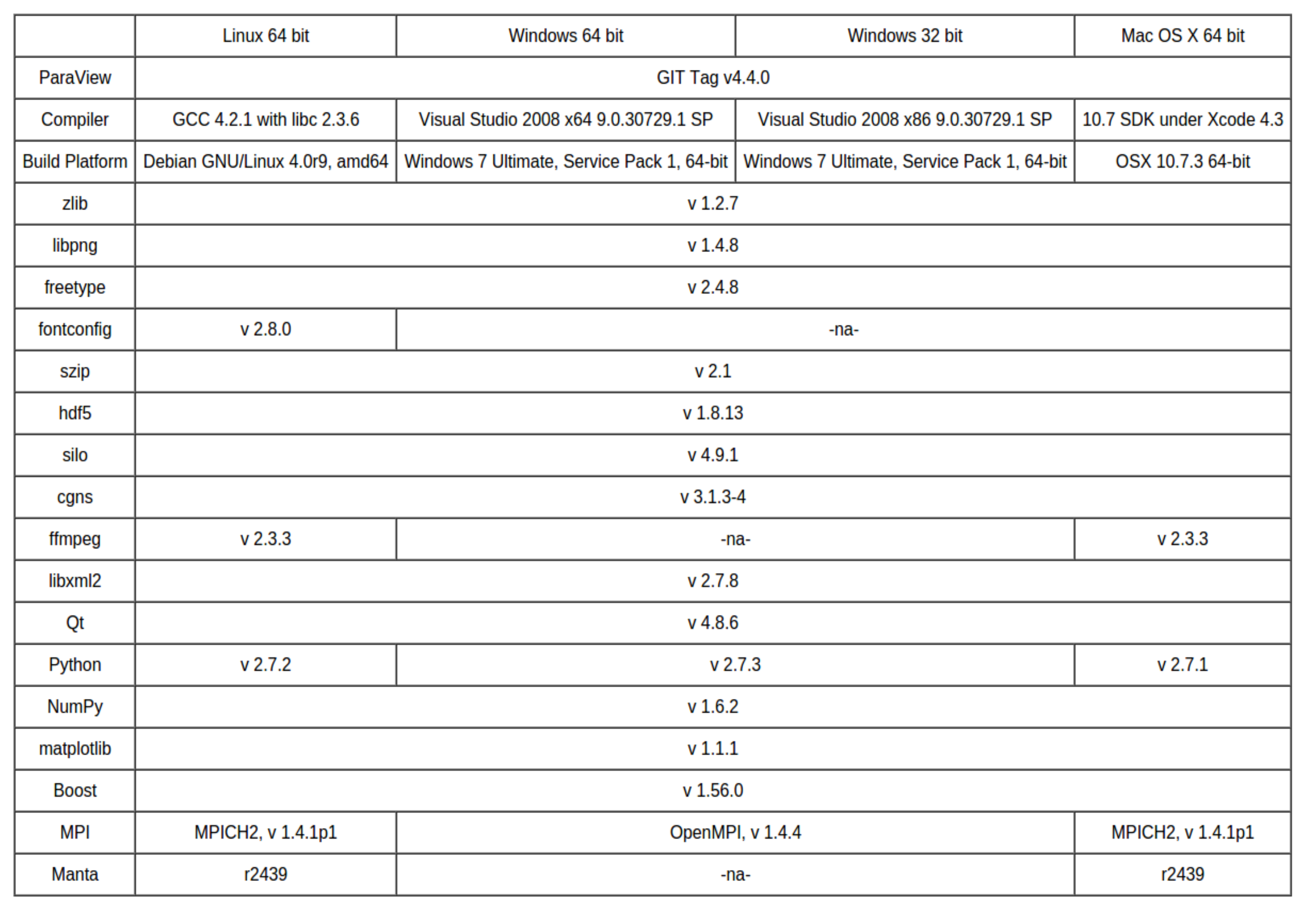
data analysis & vis





ParaView *is* VTK

dependencies



<table>
<tr><th></th><th>Linux 64 bit</th><th>Windows 64 bit</th><th>Windows 32 bit</th><th>Mac OS X 64 bit</th></tr>
<tr><td>ParaView</td><td colspan="4">GIT Tag v4.4.0</td></tr>
<tr><td>Compiler</td><td>GCC 4.2.1 with libc 2.3.6</td><td>Visual Studio 2008 x64 9.0.30729.1 SP</td><td>Visual Studio 2008 x86 9.0.30729.1 SP</td><td>10.7 SDK under Xcode 4.3</td></tr>
<tr><td>Build Platform</td><td>Debian GNU/Linux 4.0r9, amd64</td><td>Windows 7 Ultimate, Service Pack 1, 64-bit</td><td>Windows 7 Ultimate, Service Pack 1, 64-bit</td><td>OSX 10.7.3 64-bit</td></tr>
<tr><td>zlib</td><td colspan="4">v 1.2.7</td></tr>
<tr><td>libpng</td><td colspan="4">v 1.4.8</td></tr>
<tr><td>freetype</td><td colspan="4">v 2.4.8</td></tr>
<tr><td>fontconfig</td><td>v 2.8.0</td><td colspan="3">-na-</td></tr>
<tr><td>szip</td><td colspan="4">v 2.1</td></tr>
<tr><td>hdf5</td><td colspan="4">v 1.8.13</td></tr>
<tr><td>silo</td><td colspan="4">v 4.9.1</td></tr>
<tr><td>cgns</td><td colspan="4">v 3.1.3-4</td></tr>
<tr><td>ffmpeg</td><td>v 2.3.3</td><td colspan="2">-na-</td><td>v 2.3.3</td></tr>
<tr><td>libxml2</td><td colspan="4">v 2.7.8</td></tr>
<tr><td>Qt</td><td colspan="4">v 4.8.6</td></tr>
<tr><td>Python</td><td>v 2.7.2</td><td colspan="2">v 2.7.3</td><td>v 2.7.1</td></tr>
<tr><td>NumPy</td><td colspan="4">v 1.6.2</td></tr>
<tr><td>matplotlib</td><td colspan="4">v 1.1.1</td></tr>
<tr><td>Boost</td><td colspan="4">v 1.56.0</td></tr>
<tr><td>MPI</td><td>MPICH2, v 1.4.1p1</td><td colspan="2">OpenMPI, v 1.4.4</td><td>MPICH2, v 1.4.1p1</td></tr>
<tr><td>Manta</td><td>r2439</td><td colspan="2">-na-</td><td>r2439</td></tr>
</table>

highlights

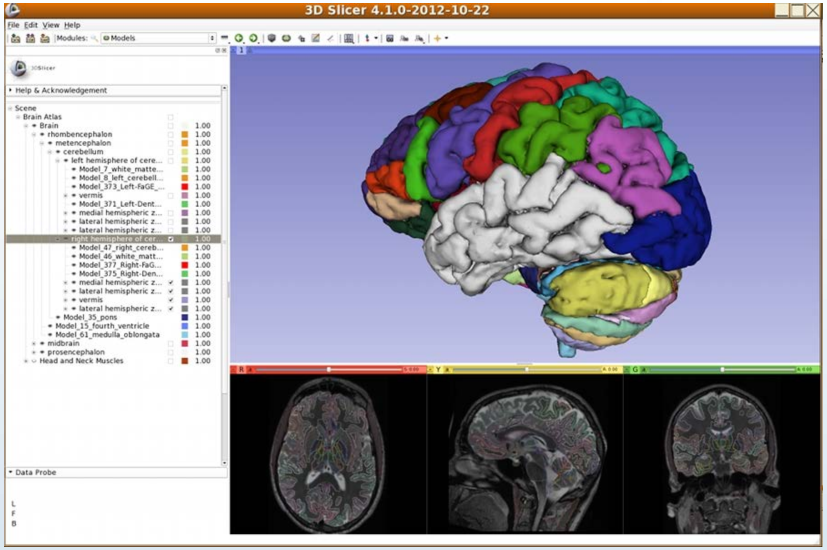
- ~60K lines of CMake code (excluding ~160K lines of CMake code from VTK)
- Uses superbuild for building dependencies, generating redistributable packages, and cross compiling.
- Broad set of dependencies: Qt, Python (NumPy, Matplotlib), Boost, MPI, CGNS, HDF5, fontconfig etc.
- Multi-platform building and packaging with CMake/CPack: OsX, Linux, Windows
- Cross compile support:
  - BlueGene Q (Mira: Argonne Leadership Computing Facility)
  - Cray XK7 (Titan: Oak Ridge Leadership Computing Facility)

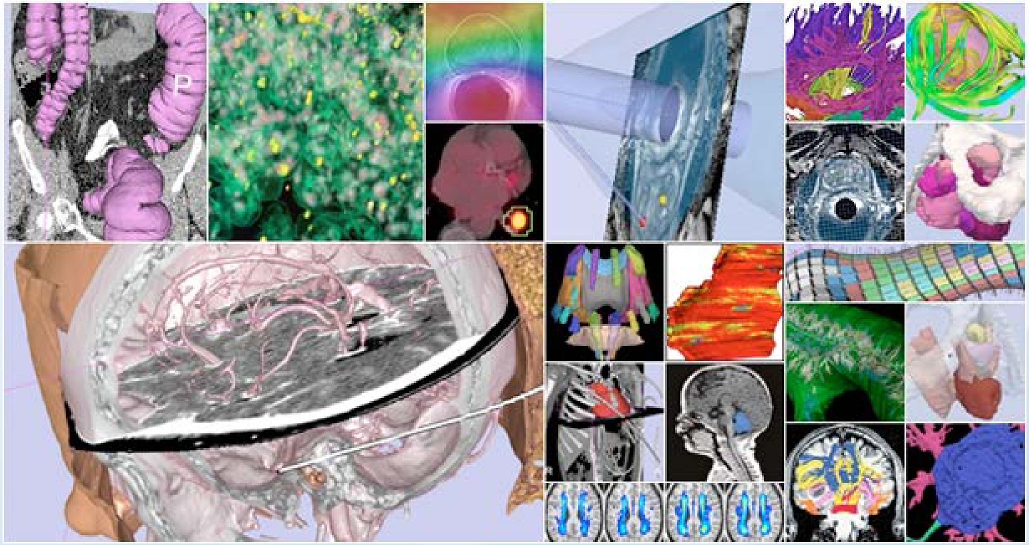
Example: 3DSlicer

An open-source platform for delivering image analysis technology for personalized medicine research.

3DSlicer

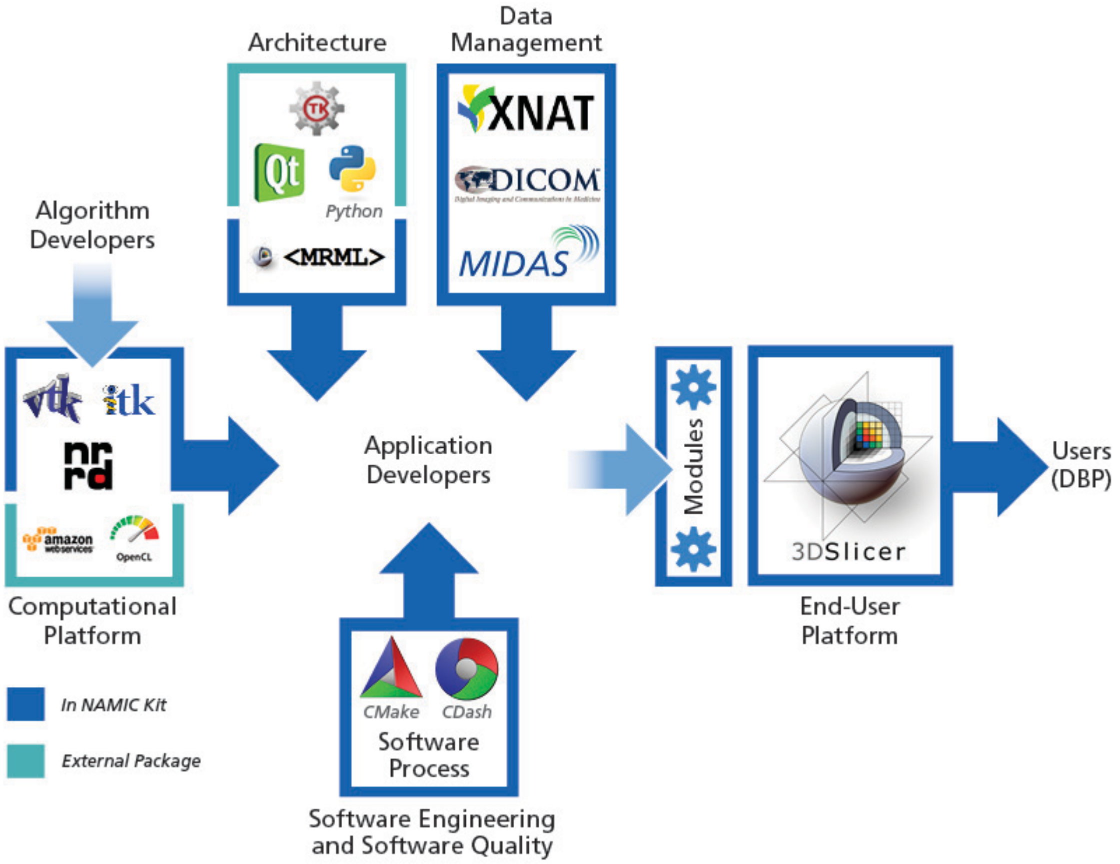
Supported by NA-MIC, NAC, BIRN, NCIGT and the Slicer Community.
See http://www.slicer.org for details.


3D Slicer 4.1.0-2012-10-22
File Edit View Help
Modules: Models
3DSlicer
Help & Acknowledgement
Scene
Brain Atlas
Brain 1.00
rhombencephalon 1.00
metencephalon 1.00
cerebellum 1.00
left hemisphere of cere... 1.00
Model_7_white_matte... 1.00
Model_8_left_cerebell... 1.00
Model_373_Left-FaGE_... 1.00
vermis 1.00
Model_371_Left-Dent... 1.00
medial hemispheric z... 1.00
lateral hemispheric z... 1.00
lateral hemispheric z... 1.00
right hemisphere of cer... 1.00
Model_47_right_cereb... 1.00
Model_46_white_matt... 1.00
Model_377_Right-FaG... 1.00
Model_375_Right-Den... 1.00
medial hemispheric z... 1.00
lateral hemispheric z... 1.00
vermis 1.00
lateral hemispheric z... 1.00
Model_35_pons 1.00
Model_15_fourth_ventricle 1.00
Model_61_medulla_oblongata 1.00
midbrain 1.00
prosencephalon 1.00
Head and Neck Muscles 1.00
Data Probe
R
Y
G



Kitware

The Slicer Ecosystem



- **Major building blocks**: VTK, ITK, CTK, Qt, Python, Teem, PythonQt, ...
- **Key infrastructure**: CMake, CTest, CDash, MediaWiki, GitHub….
- **Distribution mechanisms**: Extension manager for software distribution, data store for data distribution
- **Slicer for end-users**

Kitware

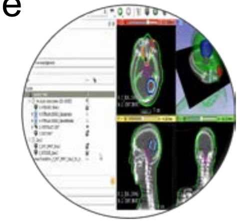
Graphic courtesy of JC Fillion-Robin, Kitware

Packages used by Slicer

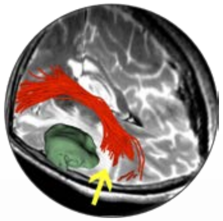
- CTKAPPLAUNCHER, CTK, CTKResEdit, curl, DCMTK, incrTcl, ITKv4, LibArchive, NUMPY, OpenIGTLink, OpenSSL, PCRE, python-async, python-chardet, python, python-gitdb, python-GitPython, python-nose, python-pydicom, python-PyGithub, python-setuptools, python-smmap, qRestAPI, Qt4, SimpleITK, SlicerExecutionModel, Swig, tcl, teem, tk, VTKv6, zlib

Examples of Clinical Projects

Radiation dose calculations

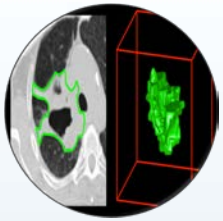


Tracking peritumoral white matter fibers



Prostate procedures

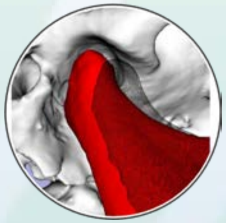
Diagnosis of Different Tumors in Lung Cancer



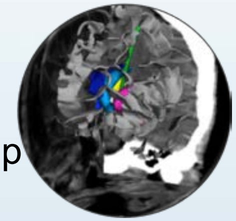
Clinical users drive creation of technology

Breast cancer surgery guidance

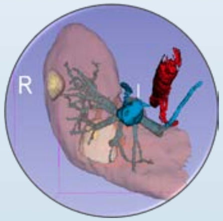
Diagnosis of Osteoarthritis Degeneration



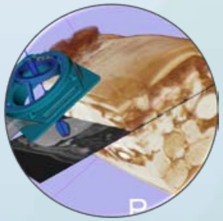
Model-Guided Deep Brain Simulation



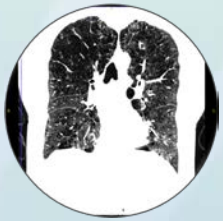
Surgical navigation



Liver procedures



Quantitative assessment of COPD



Slide courtesy of L. Cevidanes

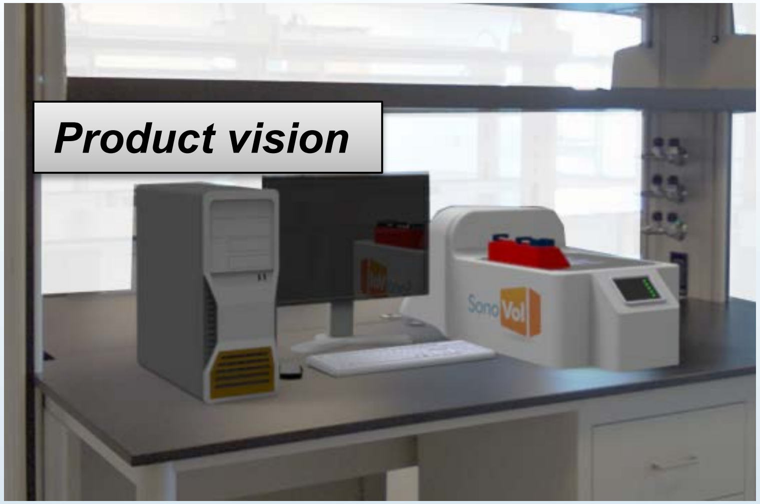
Kitware

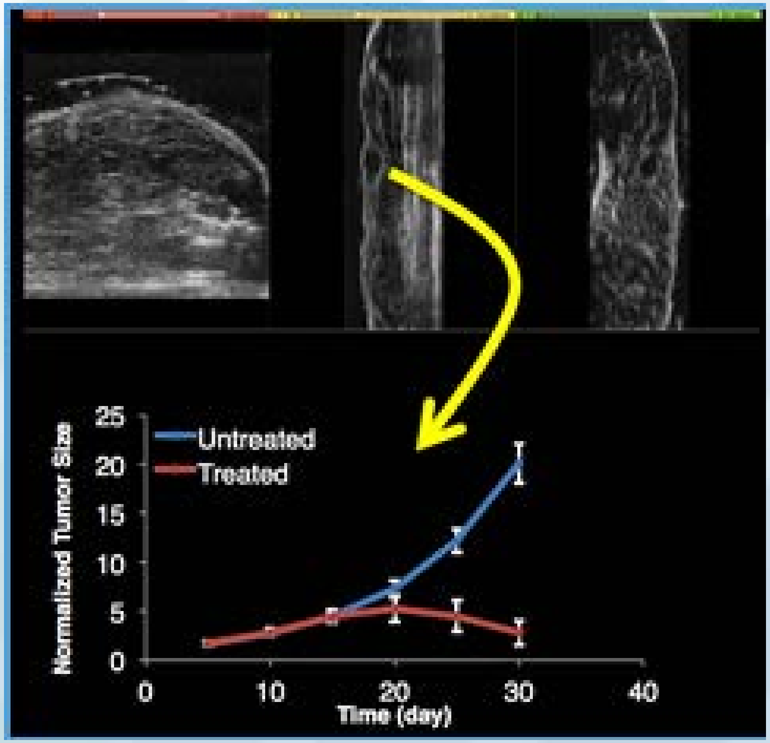
SonoVol A: "Case Study"

- SonoVol is a spinout from UNC Chapel Hill’s Joint Department of Biomedical Engineering.
- Grad student and professor designed some robotic hardware for improved preclinical ultrasound imaging
- **Needed a software interface for controlling hardware, and viewing image data.**

Ryan Gessner

Paul Dayton





- Improved “MRI-like” field of view
- Ability to study anatomical context of tumors over time
- High throughput acquisitions
- Multi-modality registration
- Improved blood vessel quantification

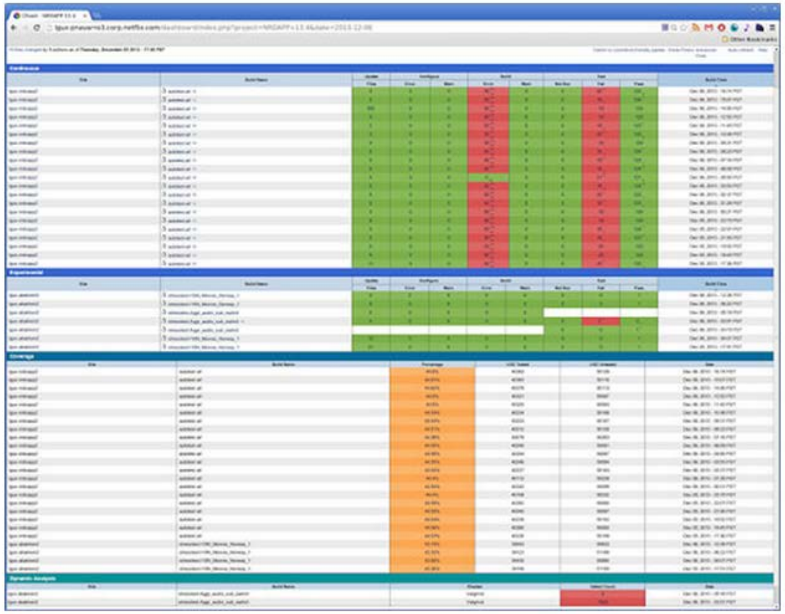
Jan
20
2014

CMake, CTest, and CDash at Netflix

Posted in CMake. Viewed 5489 times.

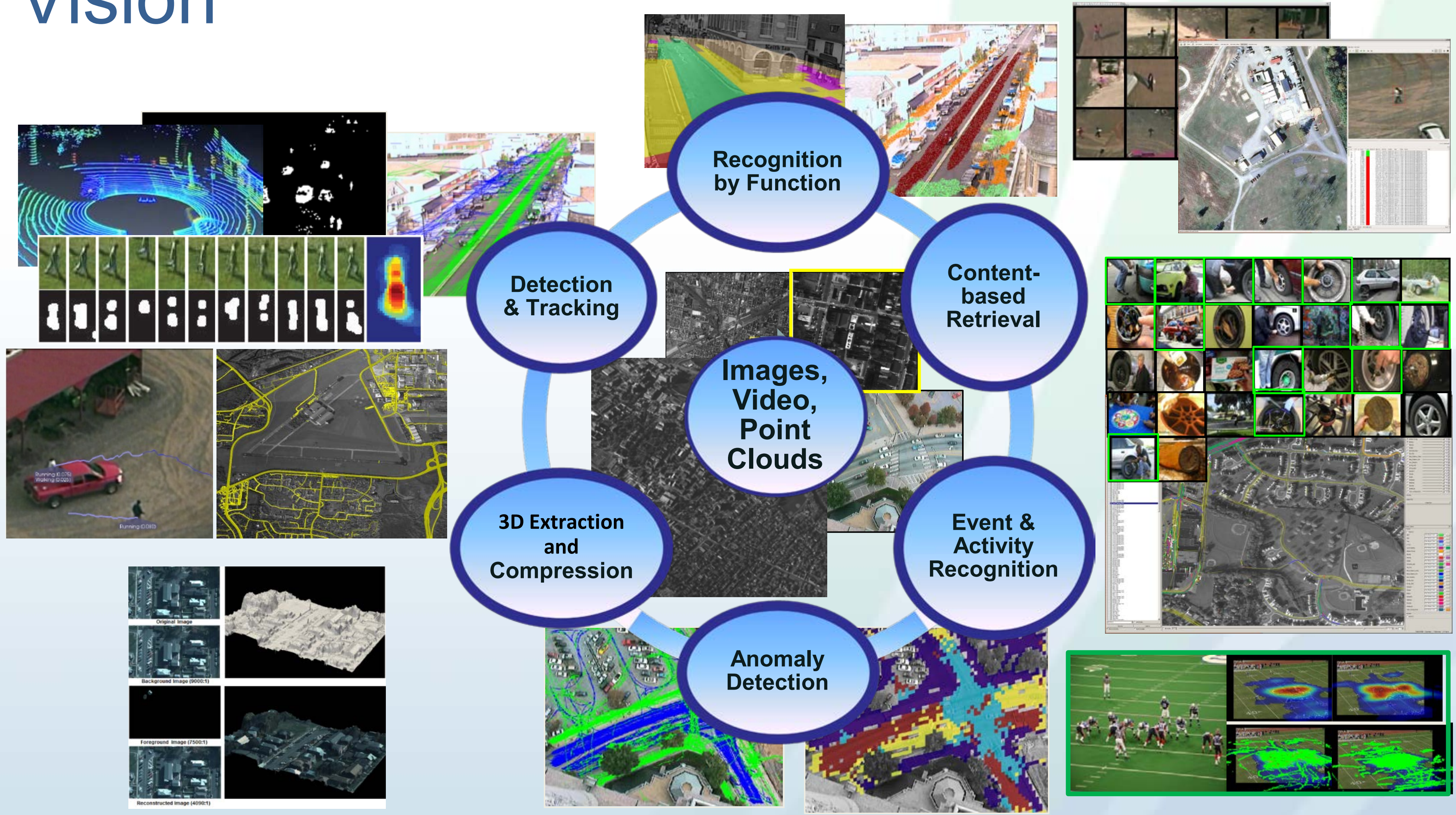
At the Core Technologies team at Netflix, we develop the application framework and streaming engine used by millions of consumer electronics devices, game consoles, tablets, and phones. With such a diverse array of devices and platforms, we need to make sure our code is lightweight, standards compliant, and portable. As we also produce the SDK that is used by partners to port Netflix to their devices, we need to make sure that it builds and runs well across many versions of the C++ compiler and standard C libraries.

We found that CMake was the tool that better fit our needs: It created project files for all development environments we used, was easily extensible with its own scripting language, provided cross-platform commands to copy and delete files and directories, and was easy to deploy on our Jenkins nodes.



Kitware Computer Vision

Object and Building Recognition by Function (DARPA)



Images, Video, Point Clouds

Recognition by Function

Content-based Retrieval

Event & Activity Recognition

Anomaly Detection

3D Extraction and Compression

Detection & Tracking

Running (0.076)
Walking (0.025)
Running (0.083)
Original Image
Background Image (9000:1)
Foreground Image (7500:1)
Reconstructed Image (4090:1)

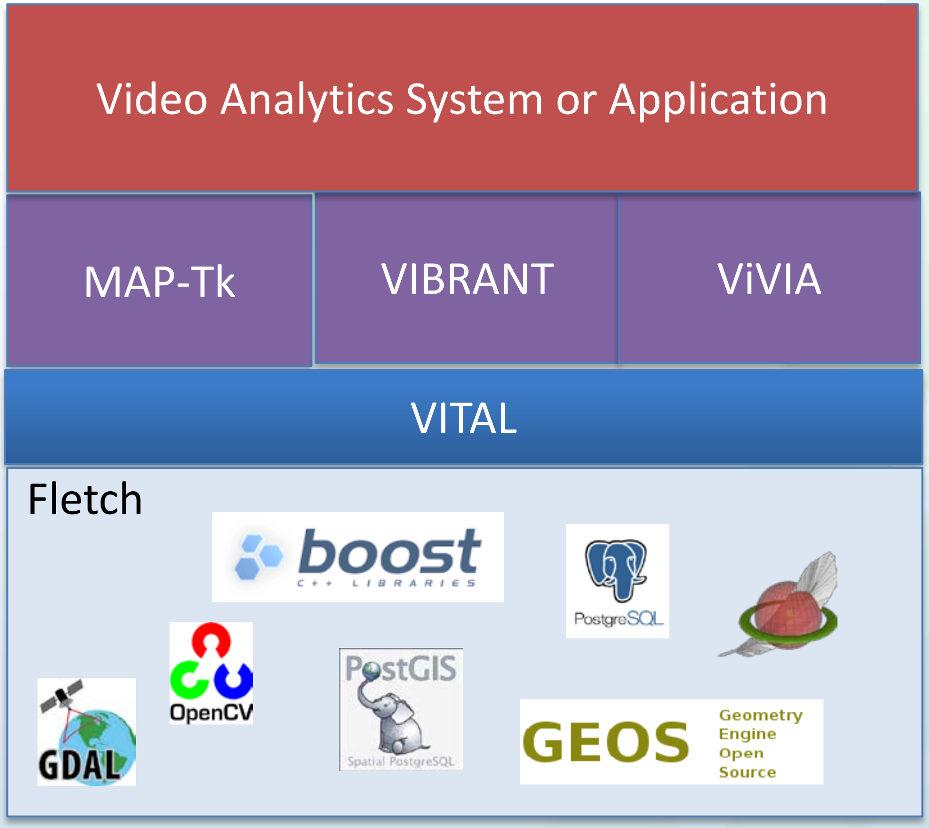
KWIVER Toolkit

Kitware Image and Video Exploitation and Retrieval Toolkit

An Open Source, production-quality video exploitation system

- Engage the community: academic, industry, government
- Avoid expensive software duplication and redundancy, speed time to solution
- Leverage the “many eyes” of the community to improve quality, stability and utility
- Bridge the gaps between research code → production software, initial feasibility → operational evaluation
- Create a true open-source community for cooperative, distributed development based upon available Open Source toolkits
- Scale down to a single researcher’s desktop and up to multi-node clusters

A KWIVER Enabled System



Video Analytics System or Application

MAP-Tk

VIBRANT

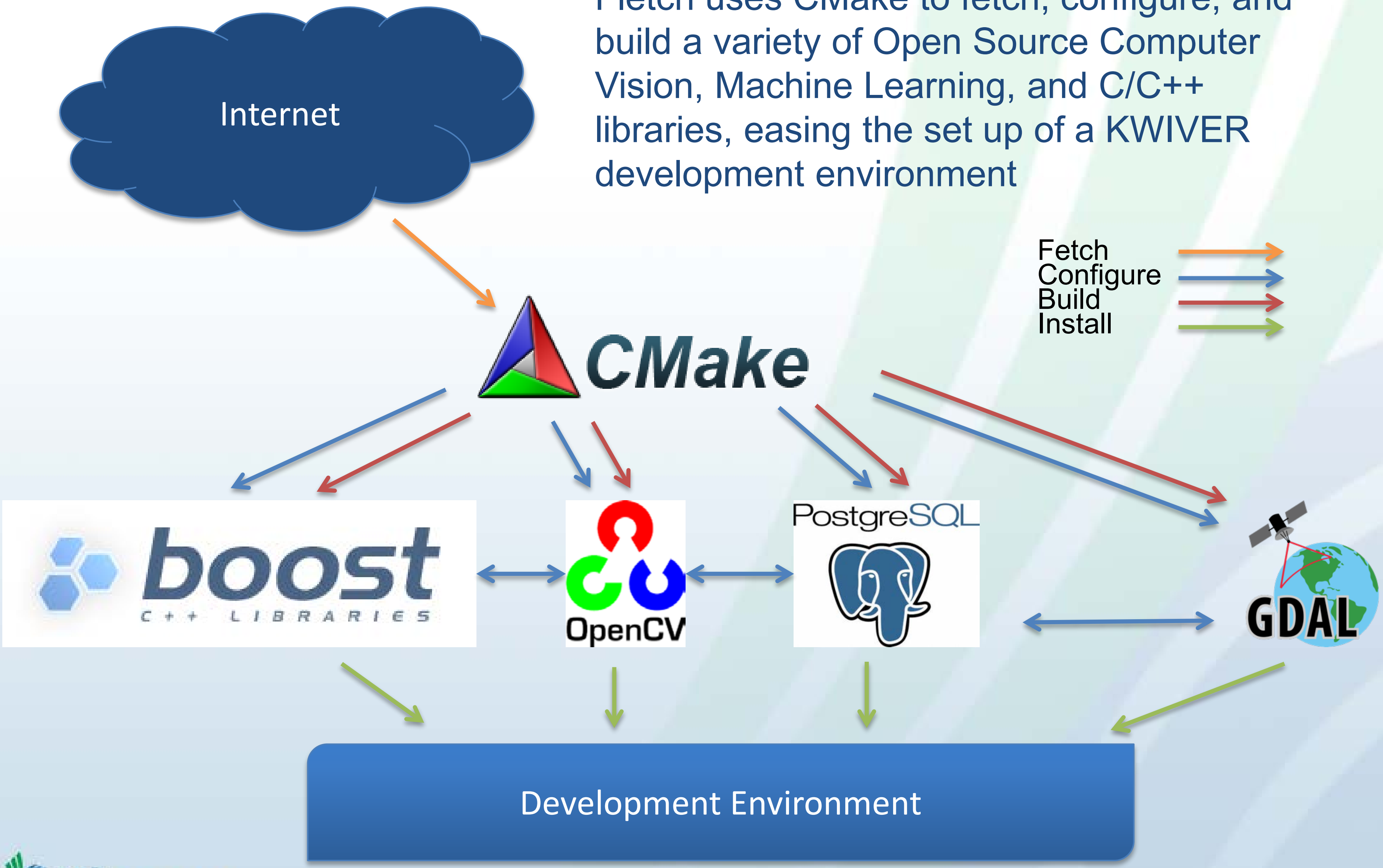
ViVIA

VITAL

Fletch

Fletch – A Computer Vision Tool Chest

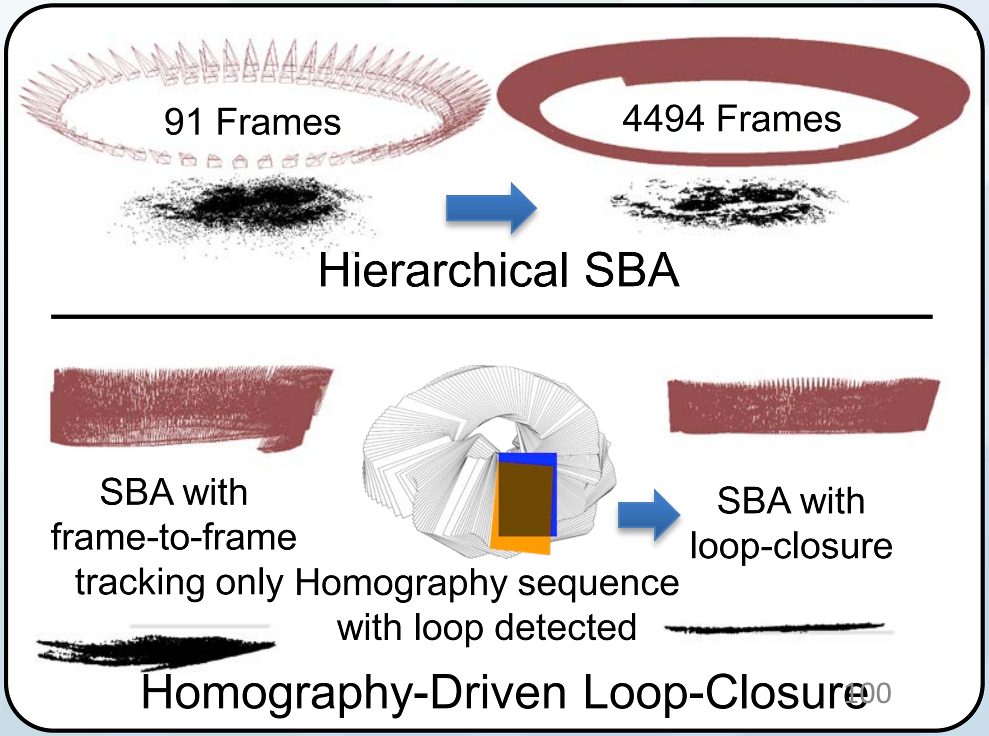
Fletch uses CMake to fetch, configure, and build a variety of Open Source Computer Vision, Machine Learning, and C/C++ libraries, easing the set up of a KWIVER development environment



MAP-Tk

Motion-imagery Aerial Photogrammetry Toolkit

- Open source with permissive BSD license https://github.com/kitware/maptk
- Highly modular, open framework
- OpenCL (GPU) accelerated feature detector and descriptor option.
- Optimized for aerial video processing
  - Frame-to-frame homography guided feature tracking
  - Homography guided loop-closure
- Recovery from bad frames during tracking
- Temporally hierarchical bundle adjustment
- Estimate shared, but unknown, intrinsics



91 Frames

4494 Frames

Hierarchical SBA

SBA with frame-to-frame tracking only

Homography sequence with loop detected

SBA with loop-closure

Homography-Driven Loop-Closure

15 Years of CMake

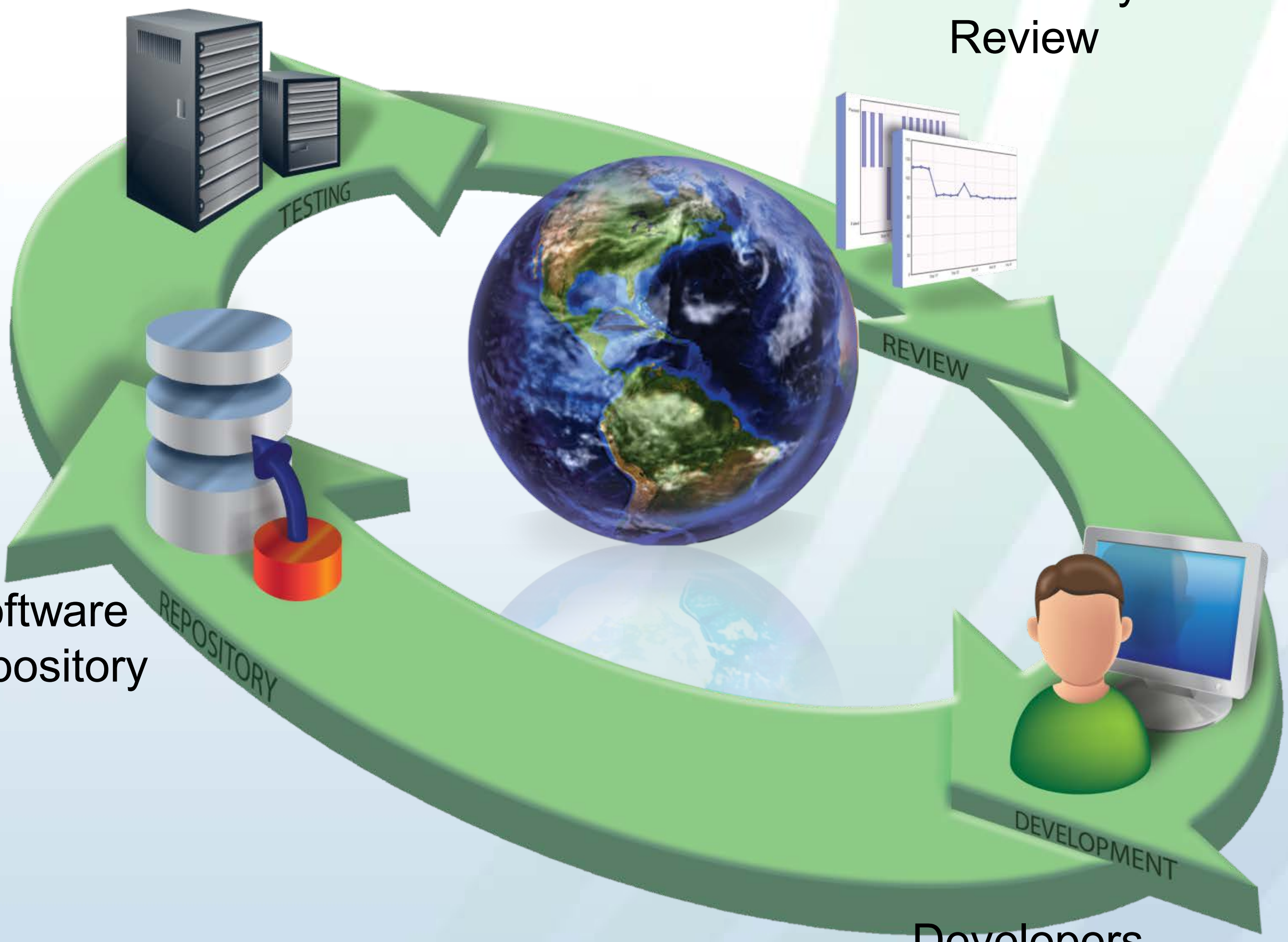
- CMake just had a birthday and is 15, over that decade and a half it has had a huge impact on scientific efforts involving C/C++

Building Science with CMake

- Reproducible builds across multiple platforms
- Integration with CI testing tools
- Integration with Data tools allowing for algorithm testing as well as unit testing
- Many domains: Medical imaging, Computer Vision, Robotics, Nuclear Energy, many others
- Open Source enabling use by many collaborators

Thanks

- Many users and contributors of CMake
- Awesome developers at Kitware
- Google and the Google Tango Project



Build, Test
& Package

Community
Review

Software
Repository

Developers
& Users