Introduction to Kitware, CMake, and Dashboards

Bill Hoffman
Kitware, Inc.
kitware@kitware.com
http://www/kitware.com
1-518-371-3971
Outline

- Nothing better to do than create useful and interesting tools
- CMake - Change the way ALL programmers build software
- Dashboards - If it's not tested it doesn't work
Kitware History

- Began writing VTK textbook in December 1993 while at GE

- Kitware Founded in 1998
  - To provide VTK support
  - Headquarters in Clifton Park NY.

- Founders: 5 previous employees of GE Corporate R&D

- Currently 37 employees; growing rapidly
  - Exceptional staff: 14 PhD, 19 MS
  - 10 new employees in the past year
  - Recently established Chapel Hill, NC office in early 2006 led by Dr. Stephen Aylward and Julien Jomier
Kitware Notable Events

- ITK Consortium 1999
  - Insight Segmentation & Registration Toolkit
  - CMake
- Tri-Lab Consortium
  - Sandia, Los Alamos, Livermore
  - ParaView, VTK for supercomputing
- National Center for Biomedical Computing
  - National Alliance for Medical Imaging Computing - NAMIC (na-mic.org)
    - One of seven, five-year, $20 million, strategic NIH roadmap sites
- Established Chapel Hill office 2006
Kitware Core Technologies

- Quality
- Accountability
- Flexibility
- Portability
- Integration

Visualization

- Large data
- Parallel computing
- Client / Server
- Web / grid architectures
- Human/Computer Interaction

Medical Image Analysis

- Registration
- Segmentation
- Image Processing
- Measurement
- Responsive GUI

Software Process

- Quality
- Accountability
- Flexibility
- Portability
- Integration
Useful and Interesting Open Source Tools from Kitware

- **VTK - The Visualization Toolkit**
  - Wrapped C++ into TCL, Python, Java
- **ParaView**
  - ParaView3 code name ParaQ in Qt
- **ITK - Insight Toolkit**
- **CMake, CTest, and CPack**
- **GCC_XML**
- **KWStyle**
- **IGSTK: Image-Guided Surgery Toolkit**
Commercial Products

- Services: books, training, support, consulting
- **VolView**: medical / scientific volume visualization
- **ParaView**: enterprise-wide visualization solutions
- **ActiViz**: COM/C# bundled visualization classes
- **MeshViz**: higher order interpolation adapters
CMake - changing the way ALL programmers compile code

- 1999 Start of the ITK project
- Kitware tasked with build environment
  - VTK current build system
    - autoconf on UNIX
    - pcmake on Windows
    - Very specific to VTK and hard to maintain, no system introspection on windows
  - CMake born out of necessity
Kitware CMake Team

Bill Hoffman  Ken Martin  Brad King  Andy Cedilnik
CMake Requirements

- Support UNIX/LINUX/Windows/MAC
- Simple easy to use
- Support code generation at build time (c++ wrapper support)
- Do not require any software other than CMake to be installed.
  - Need only a C++ compiler (same as ITK itself)
- Use Native Build tools like KDevelop, Visual Studio and Xcode
CMake: Example Usage

Windows GUI

User selects

MSVC Project

.NET Solution

Unix GUI

CMakeLists.txt

Makefile
CMake GUI (ccmake CMakeSetup)

Process Flow

- Configure
- Write CMakeCache.txt
- Did Cache Change?
  - Yes
  - No
- Generate Makefile or Project
CMake Language

• Easy for smaller projects to use larger projects

project(SampleProject)
find_package(VTK REQUIRED)
include(${VTK_USE_FILE})

set(SOURCES mySource1.cxx mySource2.cxx)
add_executable(myExecutable ${SOURCES})
target_link_libraries(myExecutable vtkRendering)
**CMake Architecture  OO C++**

- Commands are C++ objects
- Generators are C++ objects
  - Makefile, KDevelop, Xcode, MS VS, NMake, Borland Make,
- Complicated stuff should happen in the C++ code.

```
<table>
<thead>
<tr>
<th>cmCommand</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmAddLibraryCommand</td>
</tr>
<tr>
<td>cmAddTestCommand</td>
</tr>
<tr>
<td>cmGlobalGenerator</td>
</tr>
<tr>
<td>cmLocalGenerator</td>
</tr>
<tr>
<td>cmGlobalKdevelopGenerator</td>
</tr>
<tr>
<td>cmLocalKdevelopGenerator</td>
</tr>
</tbody>
</table>
```
Software Process Dashboards

CVS/SVN maintains source code revisions

CMake/CTest compiles and tests source code

Central DART Server

Results posted on web (i.e., the dashboard)

Distributed Clients

Developers review results

Developers check-in code
Dashboards -
“If it’s not tested it doesn’t work”

- Code Dashboards and our Software Process
  - Roots in GE Six Sigma initiative
- Nightly regression testing (see VTK Dashboard)
- Memory (Purify/ValGrind) testing
- Coverage testing
- Documentation evaluation
- Cross platform testing
- Nightly, Experimental, and Continuous
VTK Quality Dashboard

- MostRecentResults-Nightly/Dashboard.html
### Nightly Builds

<table>
<thead>
<tr>
<th>Site</th>
<th>Build Name</th>
<th>Update</th>
<th>Clq</th>
<th>Build Error</th>
<th>Warn</th>
<th>Min</th>
<th>Test NotRun</th>
<th>Fail</th>
<th>Pass</th>
<th>NA</th>
<th>Min</th>
<th>Build Date</th>
<th>Submit Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>r36n11.pbm.ihost.com</td>
<td>AIX53-x1C</td>
<td>19</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2.7</td>
<td>111</td>
<td>77</td>
<td>7</td>
<td>0</td>
<td>13.1</td>
<td>Sep 22 01:03 EDT</td>
<td>Fri Sep 22 01:33 EDT 2006</td>
</tr>
<tr>
<td>65.NiNet.org</td>
<td>Debian6.3-good1</td>
<td>19</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4.2</td>
<td>111</td>
<td>77</td>
<td>7</td>
<td>0</td>
<td>12.8</td>
<td>Sep 22 04:04 CDT</td>
<td>Fri Sep 22 05:44 CDT 2006</td>
</tr>
<tr>
<td>salmon.nl.mii.gov</td>
<td>Debian6.7-good2</td>
<td>19</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>13.4</td>
<td>111</td>
<td>77</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>Sep 22 04:04 CDT</td>
<td>Fri Sep 22 05:34 CDT 2006</td>
</tr>
</tbody>
</table>

### Files changed r36n11.pbm.ihost.com – AIX53-x1C as of Sep 22 01:00 EDT

- Updated files (19)
  - Bootstrap Ray 1.10 by king
  - New system component of kwsys.

- Source (40)
  - cmake/MetaStudioGenerator ray 1.13 by king
    - BUG: Centralized generation of command line arguments in escaped form. This addresses bug #70697.
  - cmake/MessageGenerator ray 1.121 by king
    - BUG: Enabled use of EscapedForShell to properly escape custom command lines. This addresses bug #67397.

### Build Name: AIX53-x1C

**68 passed, 0 failed, 0 not run**

- Name (sort by)
  - BootstrapTest: Passed (159)
  - CMakeList: Passed (0)
  - CommandLineTest: Passed (8)
  - complex: Passed (28)
Regression Testing

- Compare generated image against standard “correct” image
  - pixel-by-pixel comparison
  - can use a threshold metric
  - adjusted for effects like dithering
  - OpenGL is rather loose about image quality
Can search for the checkin that caused the problem
Why Test Daily?

- Large code base too large for any single developer to understand
- Developers distributed around the world
- Identify problems as they occur
- Insure that object API remains unchanged
- Provide feedback to developers as they experiment with new implementations
Add testing to your CMake project

- ENABLE_TESTING()
- INCLUDE(CTest)
- ADD_TEST(name command)
- ctest -D Experimental
- Shows up on Kitware public dashboard
- See DART docs for setting up your own dashboard.
KDELIBS Dashboard


Get script from Notes
cctest –S script

You too can submit!
Kitware is a rapidly growing company with a strong commitment to open source software and to CMake.

Dashboards keep software building everywhere.
Thanks to Alexander Neundorf!