# CMake as used by the Draco and Jayenne Projects

Presented to SWIFT

2012 Feb 6

Kelly Thompson, kgt@lanl.gov

CCS-2:12-06(U)

# Outline

- What parts of the CMake does Draco/Jayenne use?
- Features with examples
- Limitations and issues
- Questions

# Draco, Jayenne and Capsaicin share the Draco Build System

- **CMake** generates build projects from instructions provided in CMakeLists.txt files.
  - Local CMakeLists + draco/config/*.cmake scripts
  - Integrated with Draco development environment (standardized .bashrc, module files, editor defaults, etc.)
- **CTest** provides a built-in testing framework
- **CDash** is a web based tool that presents projects build and test status
  - On demand access
  - Email subscription based access

# Why use CMake?

- Familiarity
- Speed
  - Full dependency tracking
  - Simple scripting language for all platforms
  - Parallel builds

Timing (seconds)

| | autoconf | CMake |
|---|---|---|
| configure | 32+28 | 4 |
| make | 103 | 25 |
| make test | 63 | 16 |

- Supports all major development platforms and development toolsets.
  - Makefiles (Windows, OS/X, Linux)
  - Eclipse CDT (Windows, OS/X, Linux)
  - XCode on OS/X
  - *Supports cross compiling (i.e.: catamount systems, roadrunner)*

# Why use CMake?

- Built-in rules for common targets
  - executables/libraries
  - C/C++/Fortran
- Automatic analysis
  - implicit dependencies (C, C++, Fortran)
  - transitive link dependencies
  - order of linker search path and RPATH.
- Auto-generated build targets
  - help targets
  - preprocessor targets
  - assembly targets

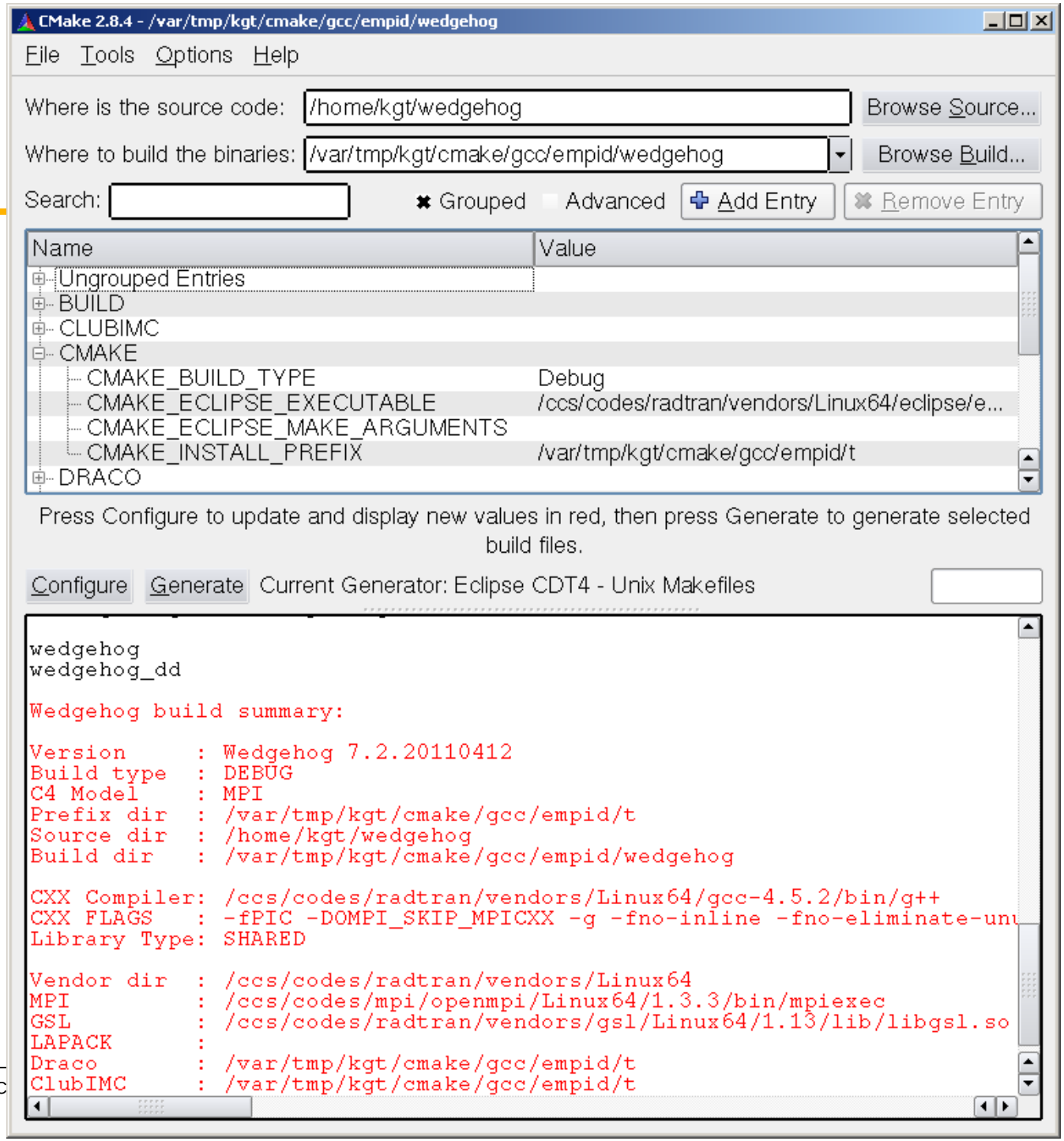
```
project( dsxx CXX )

configure_file( config.h.in
     ${PROJECT_BINARY_DIR}/ds++/config.h )

file( GLOB sources *.cc )
file( GLOB headers *.hh )

include_directories(
   ${PROJECT_SOURCE_DIR}   # sources
   ${PROJECT_BINARY_DIR} ) # config.h

add_library( Lib_dsxx "${sources}" )

install( TARGETS Lib_dsxx DESTINATION lib )
install( FILES ${headers}
     DESTINATION include/ds++ )

if( BUILD_TESTING )
   add_subdirectory( test )
endif()
```

# Run it from the command line or script

```
% cmake
   –DCMAKE_INSTALL_PREFIX=/usr/projects/draco/gcc
   –DCMAKE_BUILD_TYPE=RELEASE
   –DDRACO_DBC_LEVEL=0
   -DENABLE_RNG_NR=ON

-- The CXX compiler identification is GNU
-- Check for working CXX compiler: /usr/bin/g++
-- Check for working CXX compiler: /usr/bin/g++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info – done
-- This is DRACO version 6.1.20110412.
```

# Run it from a GUI…

- Useful when using graphical IDE.



# Why use CMake?

- Supports platform checks
- Good support for 3rd party libraries
  - FindBLAS, FindMPI, FindCUDA, …
  - Easy to implement custom Find_Package scripts: FindGSL, FindGrace
- Easy to maintain build system (lower cost than autotools).
- Generic build system is easily ported to other architectures
  - Linux, OS/X, YR, TU, CT, RR3.
- Modular build system is easily ported to other projects (e.g.: Capsaicin)
  - True for most good build systems.
- Actively supported and improvements are released at regular intervals.

# CTest is a built-in test framework

- Pass/Fail is based on return value
  - Optionally can check for regular expression match.
- Can run multiple tests in parallel
- Easy to select a subset of tests for execution
- Dependencies between tests
- Provides test time-outs
- Instrumentation:
  - memory testing via valgrind or purify
  - code coverage via gcov or BullseyeCoverage

Firefox

CDash - Draco

http://coder/cdash/index.php?project=Draco

My CDash | All Dashboards | Log Out

Tuesday, April 12 2011 10:13:04 MDT

# DRACO Dashboard

DASHBOARD | CALENDAR | PREVIOUS | CURRENT | PROJECT | ADMINISTRATION

No file changed as of **Tuesday, April 12 2011 00:10:00 MDT**

Help

[Show Filters]

## Nightly

Continuous | Experimental | Coverage | Dynamic Analysis

| Site | Build Name | Update | | Configure | | | Build | | | Test | | | | Build Time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Files | Min | Error | Warn | Min | Error | Warn | Min | NotRun | Fail | Pass | Min | |
| ccscs8 | Linux64_gcc_Debug | 4 | 0.7 | 0 | 0 | 0.5 | 0 | 0 | 3.9 | 0 | 0 | 204 | 1 | 2011-04-12T01:13:59 MDT |
| ccscs8 | Linux64_gcc_Debug_Cov | 4 | 0.9 | 0 | 0 | 0.8 | 0 | 0 | 6.7 | 0 | 0 | 204 | 3 | 2011-04-12T01:33:32 MDT |
| ccscs8 | Linux64_gcc_Release | 0 | 0.6 | 0 | 0 | 0.4 | 0 | 0 | 6.8 | 0 | 0 | 204 | 1.1 | 2011-04-12T01:03:24 MDT |
| Turing | Linux64_intel_Debug | 0 | 1.2 | 0 | 0 | 0.8 | 0 | 0 | 6.5 | 0 | 0 | 204 | 1.1 | 2011-04-12T01:05:44 MDT |
| Turing | Linux64_intel_Release | 0 | 1.2 | 0 | 0 | 0.8 | 0 | 0 | 12.8 | 0 | 0 | 204 | 0.9 | 2011-04-12T01:05:45 MDT |
| YellowRail | Linux64_pgi_Debug | 0 | 1.2 | 0 | 0 | 0.8 | 0 | 0 | 11 | 0 | 0 | 204 | 1.4 | 2011-04-12T01:05:25 MDT |
| YellowRail | Linux64_pgi_Release | 0 | 1.2 | 0 | 0 | 0.8 | 0 | 0 | 9 | 0 | 0 | 204 | 1.3 | 2011-04-12T01:05:25 MDT |
| Totals | **7 Builds** | 8 | 7 | **0** | **0** | 4.9 | **0** | **0** | 56.7 | **0** | **0** | **1428** | 9.8 | |

## No Continuous Builds

## No Experimental Builds

## Coverage

| Site | Build Name | Percentage | LOC Tested | LOC Untested | Date |
|---|---|---|---|---|---|
| ccscs8 | Linux64_gcc_Debug_Cov | 96.24% | 2328 | 91 | 2011-04-12T01:33:32 MDT |

## Dynamic Analysis

| Site | Build Name | Checker | Defect Count | Date |
|---|---|---|---|---|
| ccscs8 | Linux64_gcc_Debug | Valgrind | 26 | 2011-04-12T01:13:59 MDT |

Kitware

CDash 1.8.0 © 2010 Kitware Inc.
[report problems]

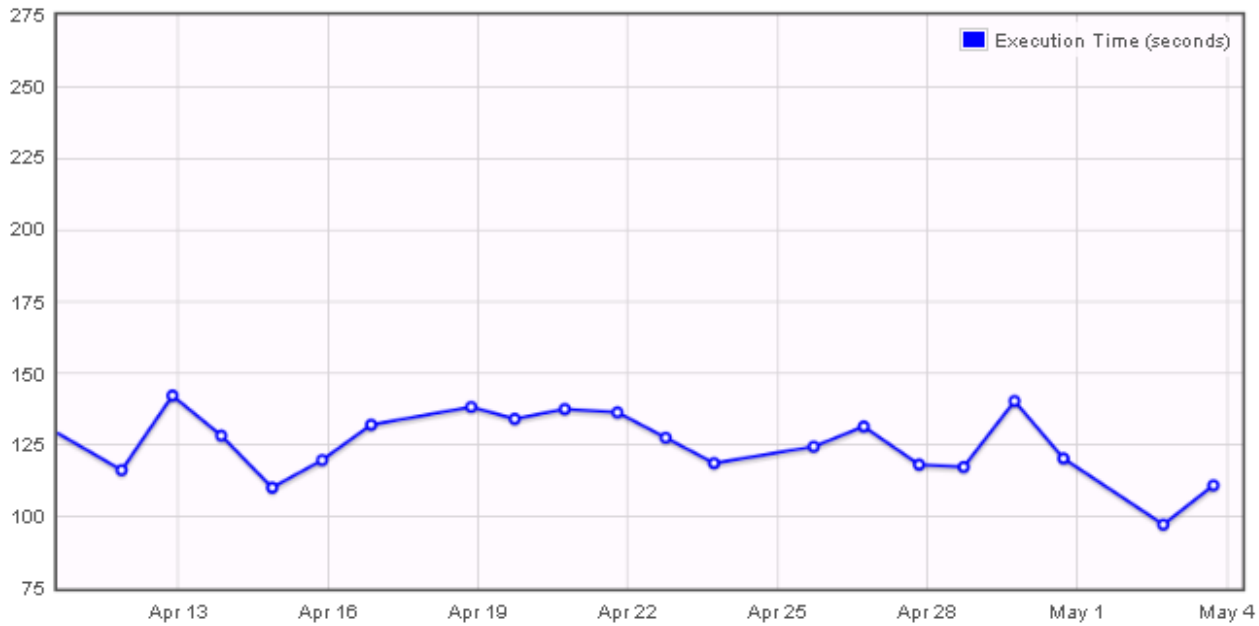
# CDash reports execution time for tests

**Site Name:** ccscs8
**Build Name:** Linux64_gcc_Debug
**Build Date:** 2011-05-03 05:23:40

milagro_xyz_python Passed

| | |
|---|---|
| **Execution Time (s)** | 110.97 (mean:115.38 std:10.83) |
| **Command Line** | /usr/bin/python2.4 "regress_milagro.py" "--procs" "2" "-x" "/home/regress/cmake_jayenne/milagro/Nightly_gcc/Debug/build/src/milagro_xyz/bin/milagro_xyz" |
| **Completion Status** | Completed |
| **Pass Reason** | Required regular expression found.Regex=[.*[Tt]est: Passed ] |

[Show Test Time Graph]
[Zoom out]




[Show Failing/Passing Graph]

# CDash reports ctest results including valgrind/purify, LOC and gcov/bullseye

Firefox

CDash : Draco

http://coder/cdash/viewCoverage.php?buildid=876&status=:

Login | All Dashboards

Tuesday, April 12 2011 10:10:44

DRACO
Dashboard

DASHBOARD BACK CALENDAR PREVIOUS CURRENT PROJECT

Coverage started on Tuesday, April 12 2011

| Coverage Summary | |
|---|---|
| Total Coverage | 96.24 |
| Tested lines | 2328 |
| Untested lines | 91 |
| Files Covered | 335 of 339 |
| Files Satisfactorily Covered | 318 |
| Files Unsatisfactorily Covered | 21 |

[Show coverage over time]

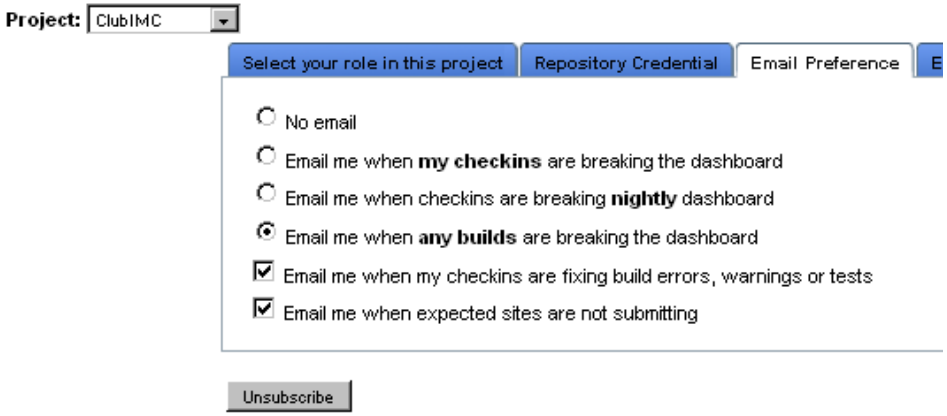
| Coverage Legend |
|---|
| Satisfactory coverage |
| Unsatisfactory coverage |
| Dangerously low coverage |

Low (5) | Medium (16) | Satisfactory (318)

| Filename | Status | Percentage | Branch Points not covered | Functions not covered | Priority |
|---|---|---|---|---|---|
| ./src/timestep/target_ts_advisor.hh | Medium | 50.00% | 0/0 | 1/2 | None |
| ./src/viz/Ensight_Stream.hh | Medium | 50.00% | 0/0 | 1/2 | None |
| ./src/ds++/conj.hh | Medium | 50.00% | 0/0 | 1/2 | None |
| ./src/timestep/ratio_ts_advisor.hh | Medium | 50.00% | 0/0 | 1/2 | None |
| ./src/ds++/Field_Traits.hh | Medium | 50.00% | 0/0 | 2/4 | None |
| ./src/ds++/Slice.hh | Medium | 50.00% | 0/0 | 13/26 | None |
| ./src/c4/scatterv.t.hh | Medium | 52.08% | 11/24 | 1/2 | None |

# CDash Email

- Subscription based
  - User's control when they should be notified.
- Simple 'click for more info' model.




**Subject: FAILED (w=1): ClubIMC - Linux64_gcc_Debug_Cov – Nightly**

A submission to CDash for the project ClubIMC has build warnings.
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://coder.lanl.gov/cdash/buildSummary.php?buildid=11730

Project: ClubIMC
Site: ccscs8
Build Name: Linux64_gcc_Debug_Cov
Build Time: 2012-02-02T02:44:10 MST
Type: Nightly
Warnings: 1

*Warnings*
src/imc/test/../Mat_State.hh line 142
(http://coder.lanl.gov/cdash/viewBuildError.php?type=1&buildid=11730)
/.../source/src/imc/test/../Mat_State.hh:142:15: warning: comparison between signed and unsigned integer expressions

-CDash on coder.lanl.gov

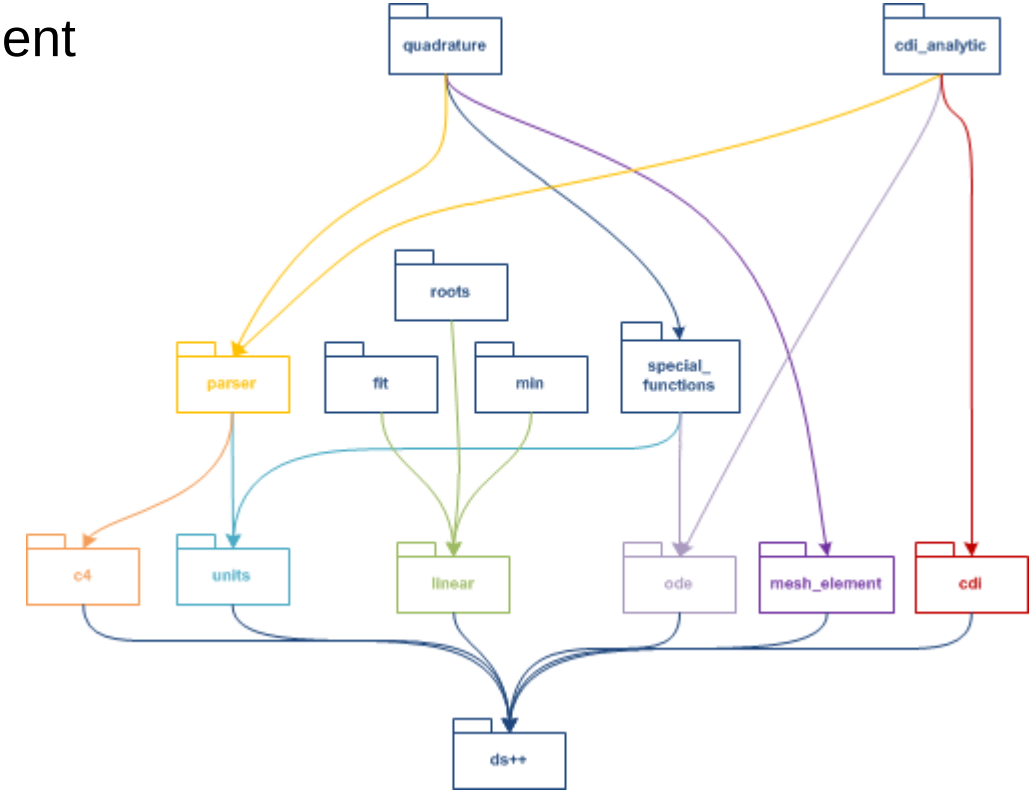
# Limitations/Issues

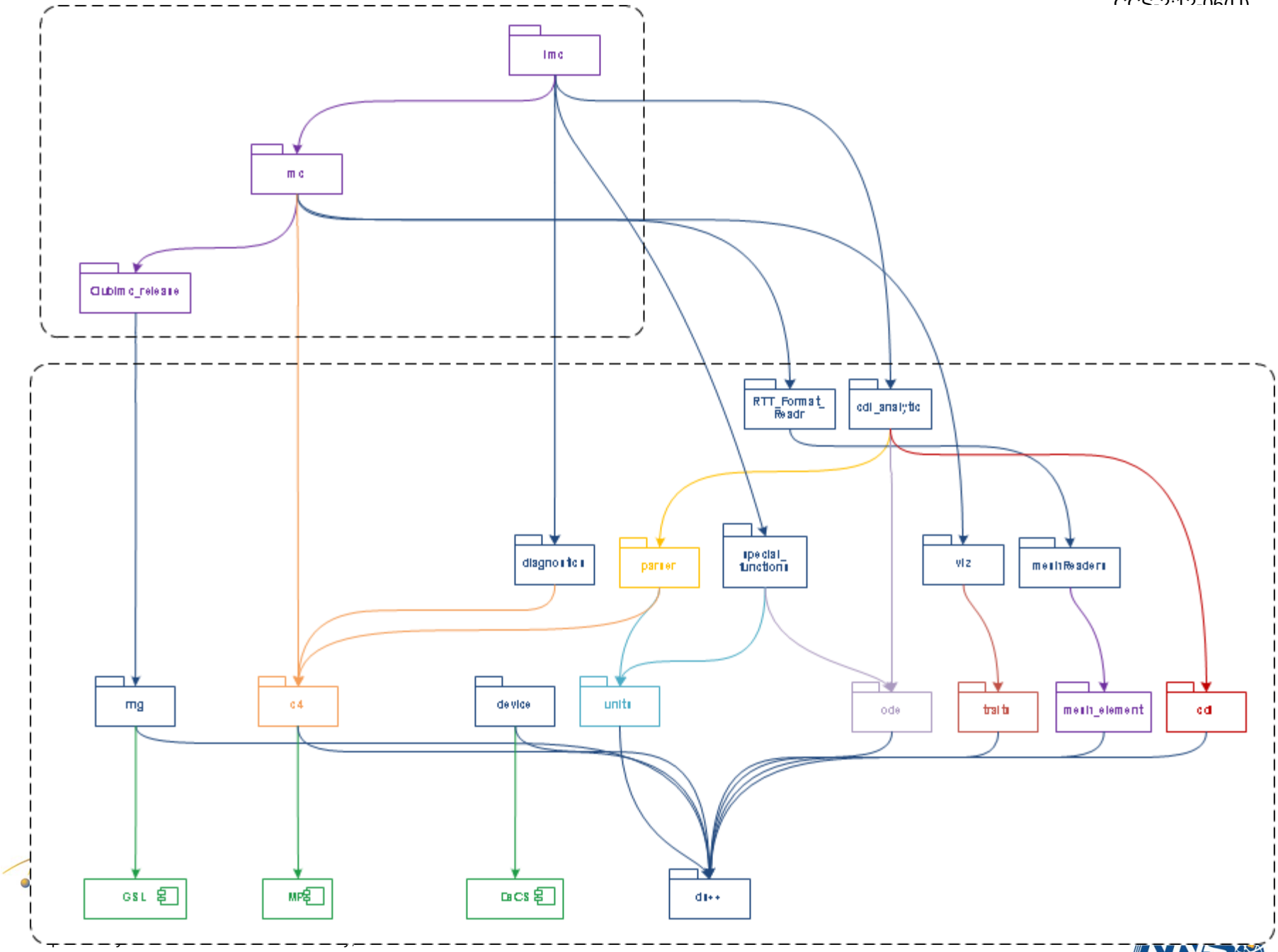
- Up-to-date versions not provided by HPC.
  - We require 2.8.6+.
- Uses its own syntax → initial learning curve.
- Limited CDash extensibility (currently cannot post plots, graphics).
  - Can attach text files (e.g.: LOC metrics).
- Difficult to micro-manage special build commands.
  - Special compile flags for a single file.
  - Working with assembly output, objdump, etc.
- The product is still young as seen by bug count and changing feature set.

# Draco build system

- Level, a-cyclic component design
- Unit testing for each component

```
# quadrature/CMakeLists.txt
file( GLOB sources *.cc )
file( GLOB headers *.hh )

include_directories(
   ${PROJECT_SOURCE_DIR}
   ${draco_src_dir_SOURCE_DIR}
   ${dsxx_BINARY_DIR}
   ${MPI_INCLUDE_PATH} )

add_component_library( Lib_quadrature
     quadrature "${sources}" )

add_dependencies( Lib_quadrature
   Lib_units
   Lib_special_functions
   Lib_ode )

if( BUILD_TESTING )
   add_subdirectory( test )
endif()
```




imc
mc
Clubimc_release
RTT_Format_Reader
cdi_analytic
diagnostics
parser
special_functions
viz
meshReaders
mg
c4
device
units
ode
traits
mesh_element
cdi
GSL
MPI
DaCS
dl++

# Draco build system

- All components are assumed to have similar design.
- CMake options are used to control system wide settings.
- We use standardized CMake target names (e.g.: Lib_quadrature)

```
#draco/config/component_macros.cmake

macro( add_component_library target_name outputname sources )

   add_library( ${target_name} ${DRACO_LIBRARY_TYPE} ${sources}  )

   if( "${DRACO_LIBRARY_TYPE}" MATCHES "SHARED" )
      set_target_properties( ${target_name} PROPERTIES
          COMPILE_DEFINITIONS BUILDING_DLL
          OUTPUT_NAME ${libraryPrefix}${outputname}
          FOLDER ${folder_name} )
    endif()

endmacro()
```

# A note on dependencies

- Build system design that focuses on common setup in each package simplifies dependency tracking.

```
#draco/src/quadrature/test/CMakeLists.txt

project( quadrature_test CXX )

set( test_deps
   Lib_parser
   Lib_c4
   Lib_special_functions
   Lib_ode
   Lib_dsxx
   ${GSL_LIBRARIES}
   ${MPI_LIBRARIES}
   ${PAPI_LIBRARY} )

add_scalar_tests(
   SOURCES "${test_sources}"
   DEPS    "${test_deps}" )
```

```
#draco/config/component_macros.cmake

macro( add_scalar_tests test_sources )

    parse_arguments(
        # prefix
        addscalartest
        # list names
        "SOURCES;DEPS;TEST_ARGS;PASS_REGEX;FAIL_REGEX;
        RESOURCE_LOCK;RUN_AFTER”
        # option names
        "NONE"
        ${ARGV} )

...
add_executable( Ut_${compname}_${testname}_exe
                ${file} )
...
add_test(...)

...
```

# Results

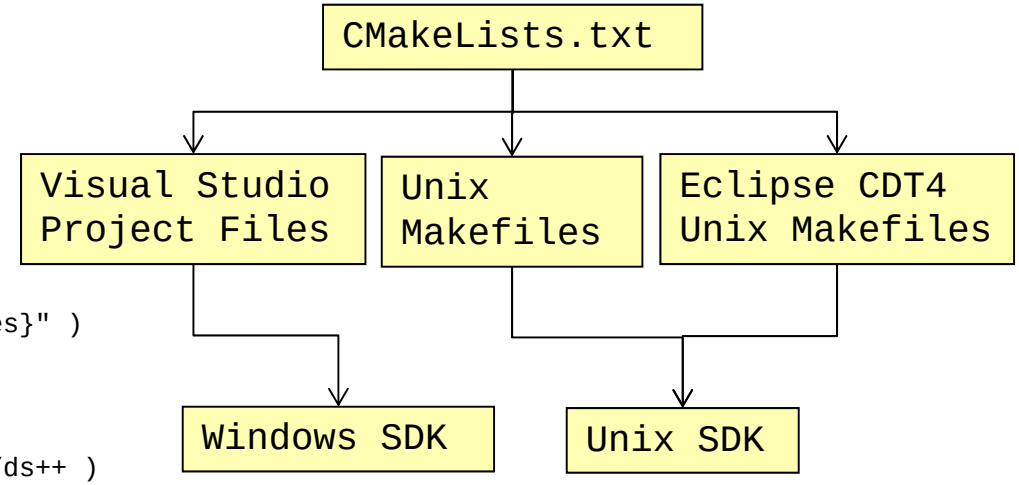
- Lower maintenance cost
  - Let someone else maintain the system when possible
  - FindVendor
  - Targets
- Multitool/multiplatform support
  - IDE or Makefiles
  - Linux, OS/X, Windows, Catamount, etc.
- Configure, build and testing speed
  - Dependencies tracking
- Associated toolset
  - CDash
  - Release/install capabilities

# Resources

- CMake, http://www.cmake.org
- Wiki, http://www.cmake.org/Wiki/CMake
- “Mastering CMake” book
- CMake Mail List Archive, http://www.cmake.org/cmake/help/mailing.html

# Samples

```
project( dsxx CXX )
configure_file( config.h.in ${PROJECT_BINARY_DIR}/ds++/config.h )
file( GLOB sources *.cc )
file( GLOB headers *.hh )

include_directories(
   ${PROJECT_SOURCE_DIR}   # sources
   ${PROJECT_BINARY_DIR} ) # config.h

add_component_library( Lib_dsxx ds++ "${sources}" )

install( TARGETS Lib_dsxx DESTINATION lib )
install( FILES ${headers} DESTINATION include/ds++ )

if( BUILD_TESTING )
   add_subdirectory( test )
endif()
```

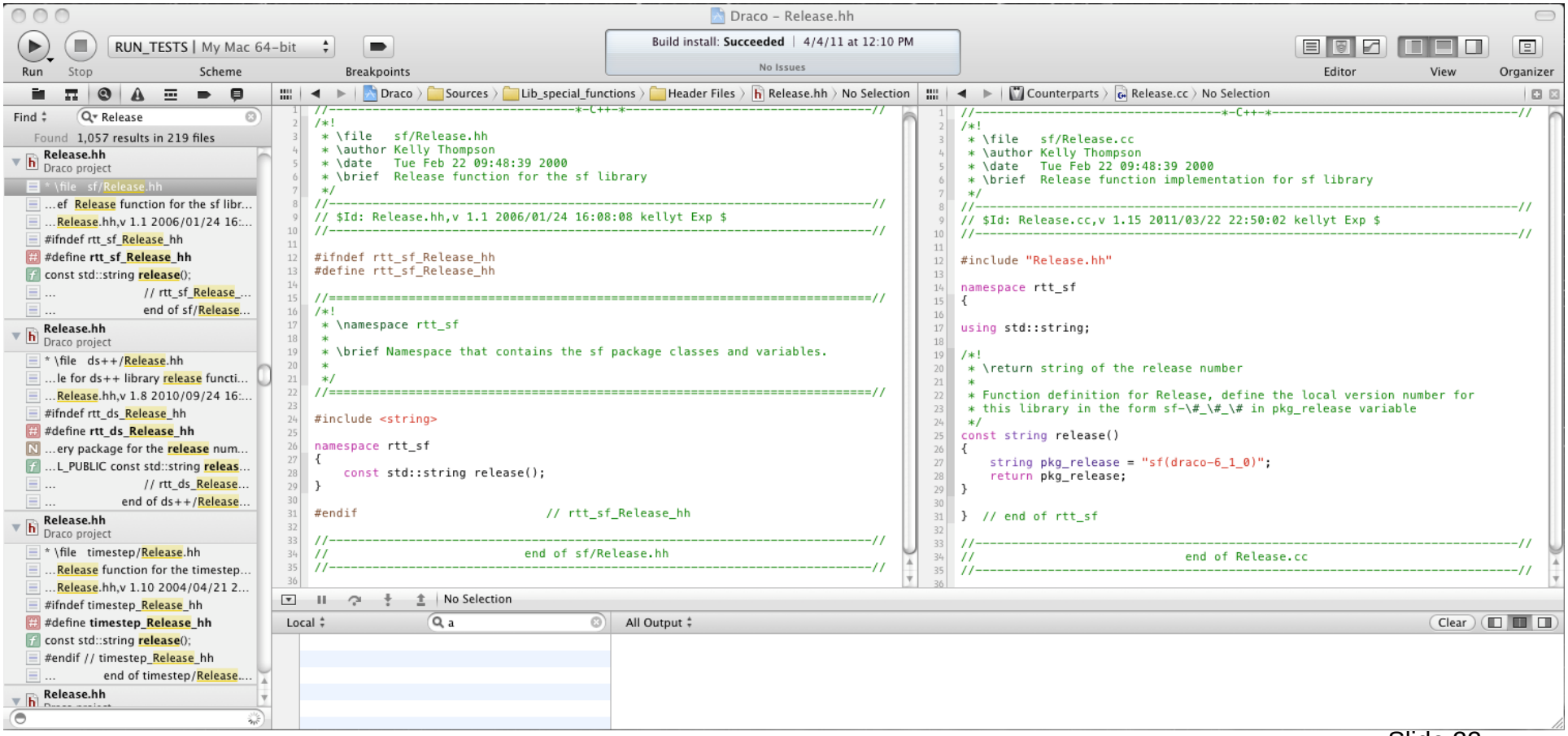


# Samples continued

```
% make
Scanning dependencies of target Lib_dsxx
[  0%] Building CXX object src/ds++/CMakeFiles/Lib_dsxx.dir/Assert.cc.o
[  0%] Building CXX object src/ds++/CMakeFiles/Lib_dsxx.dir/DynArray_pt.cc.o
[  1%] Building CXX object src/ds++/CMakeFiles/Lib_dsxx.dir/File_Streams.cc.o
[  1%] Building CXX object src/ds++/CMakeFiles/Lib_dsxx.dir/Release.cc.o
[  1%] Building CXX object src/ds++/CMakeFiles/Lib_dsxx.dir/ScalarUnitTest.cc.o
[  1%] Building CXX object src/ds++/CMakeFiles/Lib_dsxx.dir/UnitTest.cc.o
[  2%] Building CXX object src/ds++/CMakeFiles/Lib_dsxx.dir/isFinite_pt.cc.o
[  2%] Building CXX object src/ds++/CMakeFiles/Lib_dsxx.dir/Slice_pt.cc.o
[  2%] Building CXX object src/ds++/CMakeFiles/Lib_dsxx.dir/Homogeneous_New.cc.o
[  2%] Building CXX object src/ds++/CMakeFiles/Lib_dsxx.dir/dbc_pt.cc.o
[  3%] Building CXX object src/ds++/CMakeFiles/Lib_dsxx.dir/to_string_pt.cc.o
Linking CXX shared library librtt_ds++.so
[  3%] Built target Lib_dsxx
Scanning dependencies of target Lib_dsxx_test
[  3%] Building CXX object src/ds++/test/CMakeFiles/Lib_dsxx_test.dir/ds_test.cc.o
Linking CXX shared library librtt_ds++_test.so
[  3%] Built target Lib_dsxx_test
Scanning dependencies of target Ut_dsxx_tstAllocators_exe
[  3%] Building CXX object
    src/ds++/test/CMakeFiles/Ut_dsxx_tstAllocators_exe.dir/tstAllocators.cc.o
Linking CXX executable tstAllocators
```

# Samples continued

```
% cmake -G Xcode ../../draco; open Draco.xcodeproj
```



# Valgrind

Firefox

CDash : Draco

http://coder/cdash/viewDynamicAnalysis.php?

cpack

Login | All Dashboards

Tuesday, April 12 2011 13:10:30

DRACO Dashboa

DASHBOARD BACK CALENDAR PREVIOUS CURRENT

**Site Name:** ccscs8
**Build Name:** Linux64_gcc_Debug
quadrature_tQuadServices Passed

```
==29565== 80 (16 direct, 64 indirect) bytes in 1 blocks are definitely lost in los
==29565==    at 0x4A0688A: malloc (vg_replace_malloc.c:195)
==29565==    by 0x5CBA63F: gsl_permutation_alloc (init.c:36)
==29565==    by 0x4E78EE9: rtt_quadrature::QuadServices::computeD_morel() const (Qu
==29565==    by 0x4E794B8: rtt_quadrature::QuadServices::computeD() const (QuadServ
==29565==    by 0x4E7A235: rtt_quadrature::QuadServices::QuadServices(rtt_dsxx::SP<
==29565==    by 0x4157E8: test_quad_services_with_3D_S2_quad(rtt_dsxx::UnitTest&) (
==29565==    by 0x416061: main (tQuadServices.cc:1506)
```

**Dynamic analysis started on 2011-04-12 07:13:59**

**Site Name:** ccscs8
**Build Name:** Linux64_gcc_Debug

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| quadrature_tQuadServices | Passed | 5 | | | | | | | | |
| dsxx_tstSafe_Ptr | Passed | 5 | | | | | | | | |
| dsxx_tstDBC_Ptr | Passed | 5 | | | | | | | | |
| parser_tstAbstract_Class_Parser_2 | Passed | 4 | | | | | | | | |
| dsxx_tstDynArray | Passed | | | | 2 | | | | | |
| dsxx_tstAllocators | Passed | 2 | | | 1 | | | | | |
| parser_tstAbstract_Class_Parser_1 | Passed | 2 | | | | | | | | |

Kitware

CDash 1.8.0 © 2010 Kitware Inc.
[report problems]