

# Unit tests, Integration tests Physics tests

Andrea Dotti, Gunter Folger, Pere Mato  
CERN – PH/SFT  
Geant4 workshop 2012

# Overview

- Unit tests
  - What we offer
  - What we expect from developers
- Integration tests
  - What and how the «nightlies» do for integration testing
  - How developers can better contribute
  - How developers can improve tests
- Physics tests
  - A new approach

# Unit tests



Slides by

Pere Mato

CERN – PH/SFT

# Introduction

- ❖ Unit Tests in source code tree are the only missing parts to be converted to CMake
  - ❖ After this developers have no reason to not migrate to CMake
- ❖ Driving Goals/Requirements
  - ❖ Easy of use - minimal effort developers
  - ❖ Locality - adding new tests for a package requires to add/modify a single file in the package itself
  - ❖ Minimal overhead if not interested on the tests
  - ❖ Runnable eventually as part of the nightly/continuous integration tests

# Modus Operandi

- ❖ A Unit Test is either a single `.cc` file or a directory with a fix structure (single `.cc` and `/src`, `/include` directories) in a `/test` directory
  - ❖ It would be nice if the tests do not require extra arguments and returns success if execution is successful
  - ❖ Frameworks like CppUnit or similar could be used to write them
- ❖ Developer places a single `CMakeLists.txt` file in the `/test` directory
- ❖ Developer configures with `cmake -DGEANT4_BUILD_TESTS=ON`
- ❖ Developer builds the tests with `make testXXX` or `make testXXX/fast` or `make tests` to build all of them
- ❖ Developer runs each test with `output/runtime/testXXX` or all of them using CTest with `ctest -L UnitTests`

# .../test/CMakeLists.txt

default is test\*.cc  
directories are also possible

```
GEANT4_ADD_UNIT_TESTS(testMatC*.cc testA*.cc testE*.cc testG*.cc
  INCLUDE_DIRS geometry/management/include
                global/management/include
                global/HEPRandom/include
                global/HEPNumerics/include
                global/HEPGeometry/include
                materials/include
                intercoms/include
                ${CLHEP_INCLUDE_DIRS}
  LIBRARIES G4geometry)
```

dependent libraries added  
automatically

unfortunately dependent include  
directories are needed explicitly

# Build-Run session

```
[../G4/build]% cmake -DGEANT4_BUILD_TESTS=ON [-DGEANT4_ENABLE_TESTING=ON].
```

```
-- The following Geant4 features are enabled:
```

```
GEANT4_BUILD_TESTS: Build all the tests of the project
```

```
GEANT4_ENABLE_TESTING: Enable and define all the tests of the project
```

```
GEANT4_USE_SYSTEM_EXPAT: Use system EXPAT library
```

```
-- Configuring done
```

```
-- Generating done
```

```
-- Build files have been written to: /Users/mato/Development/G4/build
```

```
[../G4/build]% make testMatComponents
```

```
[ 15%] Built target G4clhep
```

```
[ 23%] Built target G4global
```

```
[ 30%] Built target G4intercoms
```

```
[ 38%] Built target G4materials
```

```
[ 46%] Built target G4graphics_reps
```

```
[100%] Built target G4geometry
```

```
[100%] Building CXX object source/materials/test/CMakeFiles/testMatComponents.dir/testMatComponents.cc.o
```

```
Linking CXX executable ../..../outputs/runtime/testMatComponents
```

```
[100%] Built target testMatComponents
```

```
[../G4/build]% outputs/runtime/testMatComponents
```

```
***** Table : Nb of materials = 4 *****
```

```
[../G4/build]% ctest -L UnitTests
```

```
Test project /Users/mato/Development/G4/build
```

```
Start 1: testProElectroMagField 1/54
```

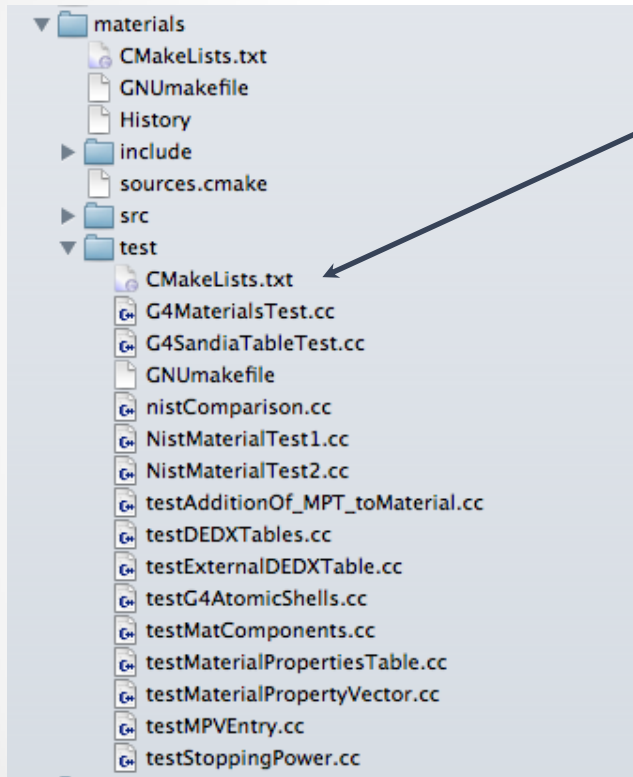
```
Test #1: testProElectroMagField ..... Passed 1.57 sec
```

```
Start 2: testPropagateMagField 2/54
```

```
Test #2: testPropagateMagField ..... Passed 0.67 sec
```

```
Gunter Folger - CERN PH/SFT
```

# Summary



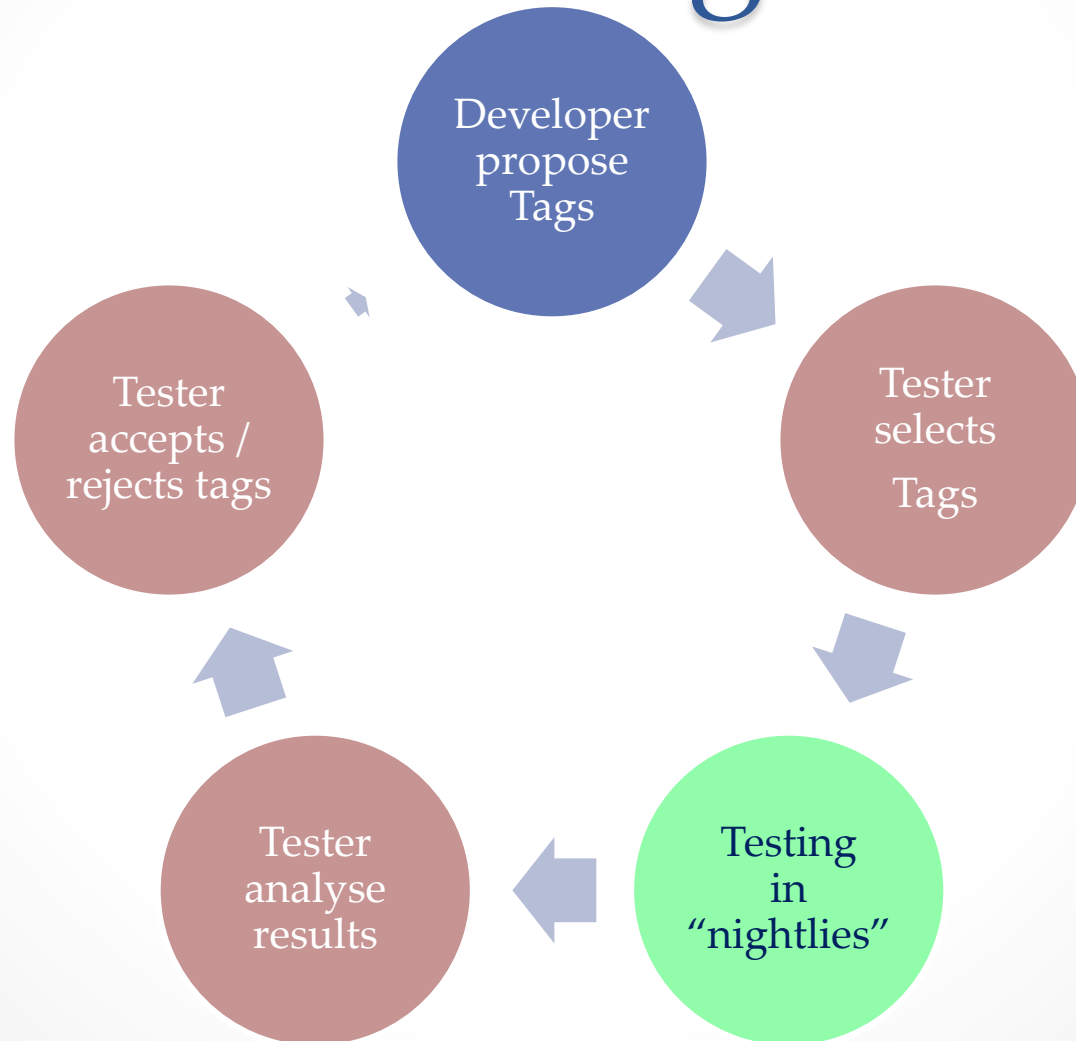
- ❖ Add CMakeLists.txt with the lists of tests (wildcards allowed)
- ❖ Enable some CMake options
- ❖ Build and Run as usual
  - ❖ build from the binary directory
  - ❖ input files may stay in source directory or copied to binary directory
- ❖ We should encourage developers to add more tests



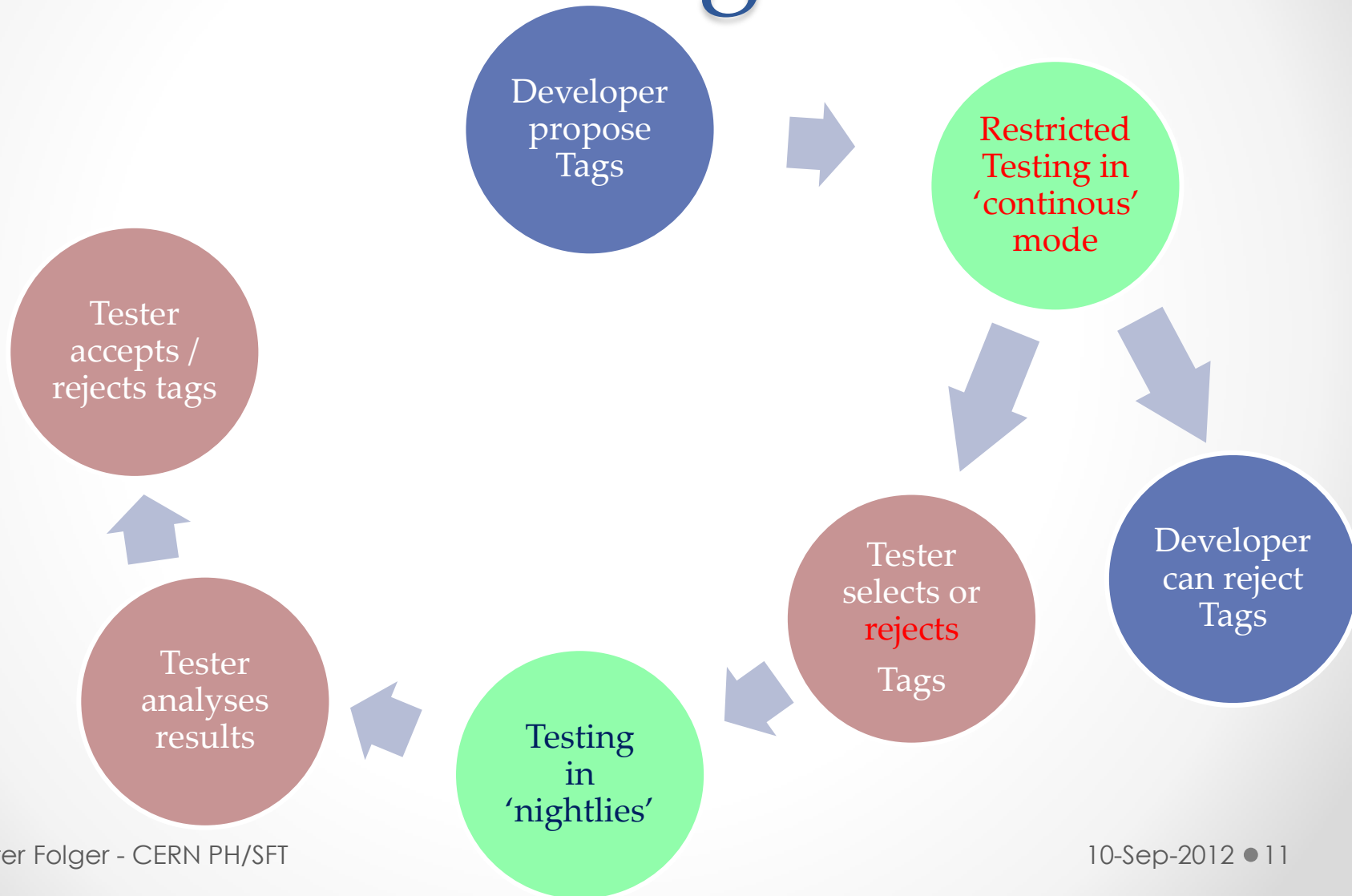
# Integration tests

...

# Regular integration testing



# Regular integration testing



# What is tested ... and how

- Testing migrated to CMake/CTest/Cdash
  - Ctest/Cdash offer testing in multiple build groups
- Continuous build group
  - geant4/tests, except few which are excluded
- Nightly build group
  - geant4/tests, except few which are excluded
  - Many geant4/examples
  - benchmarks/calorimeter/FullCMS for main physics lists
  - benchmarks/calorimeter/HadCalCMS with all physics lists
  - Full list of tests is available in cdash per platform
- Criteria for success/fail of test unchanged
  - Test runs to completion
  - stderr is empty
  - Currently no other check, up to developer to implement

# How to improve testing

- Make sure errors are reported on stderr!
- Test program should check for good/expected results
  - Check relevant quantities, or physics result.
- Regression testing - session 7a
  - Compare quantities to previous accepted values
  - Tool to statistically compare histograms recently developed
  - Example for this Implemented in PhysicsTests - test73
- Plans to extend setup to include tools like valgrind
  - Checking for memory access errors
  - Checking for leaks



# Physics Tests

• • •

Andrea Dotti  
CERN – PH/SFT

# Physics Monitoring

- { **Specialized CDash group** to collect automatic testing of physics results
  - { **Not used by shifter to reject/accept tags**
  - { Ideally all model developers should provide at least one of such tests
  - { Will also use **complete applications and examples** from G4 source
- { If test or example produces ROOT output
  - { **Automatic testing against reference** can be performed
  - { If **output changes** w.r.t. defined reference test FAILS
  - { Uses package available at:  
    <svn+ssh://svn.cern.ch/repos/g4tests/trunk/verification/StatTest>
  - { Could be extended to read in **different format files**

Discussion at Session Parallel 7A

# Modified CMakeLists.txt

```
#Perform simulation test: this test produces a ROOT file called result.root
```

```
GEANT4_ADD_TEST(mytest-run
```

```
    COMMAND ${CMAKE_CURRENT_BINARY_DIR}/mytest ...
```

```
    LABELS PhysicsChecks
```

```
    ENVIRONMENT ${GEANT4_TEST_ENVIRONMENT})
```

```
#Execute regression testing only if StatTest program is available
```

```
find_package(StatTest QUIET)
```

```
if(STATTEST_FOUND)
```

```
    STATTEST_ADD_TEST( mytest-checkOutput
```

```
        G4TEST mytest-run
```

```
        CONFIG testconf.qa
```

```
        INPUT result.root
```


```
        REFERENCE referencefile.root
```

```
        IMG imagefile.pdf
```

```
        LABELS PhysicsChecks)
```

```
endif()
```

Regression Configuration  
see StatTest/example/testconf.qa  
Web page in preparation





# Example

```
97: Test command: /Applications/CMake\ 2.8-8.app/Contents/bin/cmake "--DCMD=/usr/bin/python#/Users/adotti/workspace/gea
=/Users/adotti/workspace/testami/SimplifiedCalorimeter/FTFP_BERT-FeSci" "-P" "/Users/adotti/workspace/geant4-source/cm
97: Test timeout computed to be: 1500
97: Validation of SimplifiedCalorimeter testing suite data
97: =====
97:             hLongProfile : FAILED      (Test: BinnedWeighted1DChi2Test)
97:             hTransvProfile : NOTPASSED  (Test: BinnedWeighted1DChi2Test)
97:             /Spectra/hProtonSpectrum : FAILED      (Test: Binned1DChi2Test)
97:             /Spectra/hPionZeroSpectrum : FAILED      (Test: Binned1DChi2Test)
97:             /Spectra/hPionMinusSpectrum : SUCCESS    (Test: Binned1DChi2Test)
97:             /Spectra/hPionPlusSpectrum : FAILED      (Test: Binned1DChi2Test)
97:             /Spectra/hKaonPlusSpectrum : SUCCESS    (Test: Binned1DChi2Test)
97:             /Spectra/hKaonMinusSpectrum : SUCCESS    (Test: Binned1DChi2Test)
97:             /Spectra/hKaonZeroSpectrum : SUCCESS    (Test: Binned1DChi2Test)
97:             /Spectra/hAntiProtonSpectrum : SUCCESS    (Test: Binned1DChi2Test)
97:             /Spectra/hAntiNeutronSpectrum : SUCCESS  (Test: Binned1DChi2Test)
97:             /Spectra/hHyperonsSpectrum : SUCCESS    (Test: Binned1DChi2Test)
97:             /Spectra/hNucleiSpectrum : FAILED      (Test: Binned1DChi2Test)
97:             /Spectra/hGammaSpectrum : NOTPASSED    (Test: Binned1DChi2Test)
97:             /Spectra/hNeutronSpectrum : NOTPASSED   (Test: Binned1DChi2Test)
97:             SimplifiedCalorimeter:EDEP_ACT : SUCCESS (Test: AndersonDarlingTest)
97:             SimplifiedCalorimeter:EDEP_CAL : SUCCESS (Test: AndersonDarlingTest)
97: =====
97: CMake Error at /Users/adotti/workspace/geant4-source/cmake/Modules/Geant4TestDriver.cmake:91 (message):
97:   output error: Number of tests FAILED or NOT PASSED: 8
97:
97:
97:
97: 1/1 Test #97: SimplifiedCalorimeter-FTFP_BERT-FeSci-checkOutput ...***Failed    8.34 sec
```

Histograms checks

Ntuple check

PDF file is created, can test histos or Ntuples  
few statistical tests available ( $\chi^2$ , Anderson Darling, Kolmogorov Smirnov)

# Status

- [ **Currently we are in development phase of the common tools:**
  - [ test73-\* : example of a test that fails if a fit is not as expected. **Does not use automatic testing**, all contained in classic “G4 test” style (example on where to start from)
  - [ SimplifiedCalorimeter-\* : **example of testing with an external application**. Uses automatic regression testing against reference
  - [ test30-\* (not yet in SVN): example of a test that has been **extended with the automatic regression testing**
- [ Issues/todos:
  - [ **Where to put reference files?** We do not want to “pollute” G4 SVN repository
  - [ How to update references in a simple way when we have a new Geant4 release? What should be the policy to change a reference?
  - [ Once these are solved we will put the system in production mode
- [ See: <https://twiki.cern.ch/twiki/bin/view/Geant4/PhysicsValidationTaskForce>

# Summary

- Testing is expanding
  - Load distributed to shift workers
  - More contributors to infrastructure
- CTest/CDash setup offers more flexibility
  - Simple validation can be included, physics checks.
- Developers need to use new tools and possibilities
  - After submission of tag, check result
  - Improve test code to verify correctness of test results