

Towards a testing infrastructure for the FCC SW

Lukas Marti

AEC-LHEP
University of Bern

20.10.2014

What it is about

- (Unit) tests are frequently used in SW development – help to
 - find problems
 - understand purpose of module
 - facilitate change and integration
- Running the tests automated on a nightly/daily basis:
 - shows all dependencies are correct (within FCC package but also outside)
 - helps to provide a working product at any given point
 - helps pinpoint problems as at most 24h of commits need checking
- I'm working to get this infrastructure up

Implementation of a test

- FCCSW uses CMake
- Has basic testing facilities integrated: CTest
- Use Gaudi framework:
 - Wrapper around CTest as well as testing tool QMTest and provides features to it
- Can run executable or python script, with or without additional options
- Criteria for a successful test can be:
 - Specific return code
 - Output (or lack thereof) in stdout or stderr
 - Match with a reference file
 - Customization easily possible (e.g. requiring existence of branch in a root file)
- Put information on technical implementation in the README file ([link](#))

Minimal example

- Test files are quite simple:

```
-bash-4.1$ cat ../Reconstruction/tests/qmtest/reconstruction.qms/reconstruction_min.qmt
<?xml version="1.0" ?><!DOCTYPE extension PUBLIC "-//QM/2.3/Extension//EN"
'http://www.codesourcery.com/qm/dtds/2.3/-//qm/2.3/extension//en.dtd'>
<extension class="GaudiTest.GaudiExeTest" kind="test">
<argument name="program"><text>exTest.py</text></argument>
<argument name="args"><set><text>Input.txt</text></set></argument>
</extension>
```

- Will run the python script passing Input.txt
- If return code is 0 and stderr empty the test is successful

Status

- Scheduler will be setup soon to run tests nightly
- Put a trivial test as example in my repo
- Results are sent to a public dashboard: <http://cdash.cern.ch/index.php?project=FCC>
- Triggered manually (1 line, any user can do it anytime):

The screenshot shows a web browser displaying the CDash dashboard for the FCC project. The address bar shows the URL cdash.cern.ch/index.php?project=FCC&date=2014-11-18. The page header includes navigation links like 'Dashboard', 'Calendar', 'Previous', 'Current', 'Next', and 'Project'. A status message indicates 'No file changed as of Tuesday, November 18 2014 - 02:00 CET'. The main content area is titled 'Nightly' and contains a table with build results.

Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
ui02.lhep.unibe.ch	Linux-lcg-g++-4.8.1	0	0	0	0	0	0	0	1	Nov 18, 2014 - 10:10 CET

The footer of the dashboard includes the Kitware logo and text: 'CDash 2.0.2 © Kitware | Report problems | 0.022s'.

Conclusion & Outlook

- Work on testing infrastructure is coming along nicely (thanks to Benedikt for the support)
- Skeleton is in my repo, scheduler should be ready soon
- Submitting of results to dashboard is working

Next steps:

- Include full stack in Nightly build
- Add some non-trivial examples:
 - Something like the tutorial?
 - Step by step through these modules? (Reader, ... , output)?