

# CLAS12 Offline Code Management and Validation

Nathan Harrison  
UNG

CLAS Collaboration Meeting  
March 7, 2018  
Jefferson Lab

# Outline

- Summary of existing management and validation tools
  - clas12-offline-software GitHub repository
  - Maven project management tool
  - Travis Continuous Integration (CI) testing system
- Recent improvements and updates
  - Shorter and simpler build procedure
  - Improved implementation of unit testing
  - Automatic code coverage reports
  - Automatic SpotBugs reports

- Summary of existing management and validation tools
  - clas12-offline-software GitHub repository
  - Maven project management tool
  - Travis Continuous Integration (CI) testing system

<> Code Issues 1 Pull requests 1 Projects 0 Wiki Settings Insights

Report problems

CLAS12 Offline Software

Edit

Add topics History of all changes

Frozen releases (downloads and notes)

1,191 commits

2 branches

9 releases

21 contributors

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

Different branches for parallel development

File	Commit Message	Time
bin	added bin/ etc/ and lib/ directories	3 months ago
common-tools	magfield computes greadients and ced displays	3 days ago
docs	Updated documentation	2 months ago
etc	fixed typo in EVENT.json	6 days ago
external-dependencies	Minor dependency management change	7 days ago
reconstruction	Merge branch 'master' of https://github.com/JeffersonLab/clas12-offli...	3 days ago
validation	Updated yaml and unit-tests for new EB	24 days ago
.gitignore	Cleanup	7 days ago
.travis.yml	Updated unit tests	2 months ago
README.md	Updated documentation	4 days ago
build-coatjava.sh	Minor dependency management change	7 days ago
pom.xml	fixed unstable build warnings by creating a local mvn repo to store t...	2 months ago

Source code

Maven Project Object Model file

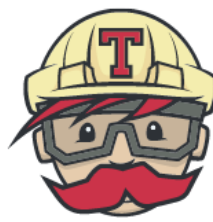
- Manages dependencies
- Compiles code
- Runs plugins
- Builds documentation (JavaDocs)

Current build status and link to Travis CI

clas12-offline-software

build passing

github.com/JeffersonLab/clas12-offline-software



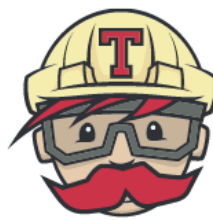
## JeffersonLab / clas12-offline-software



build passing

Current Branches Build History Pull Requests

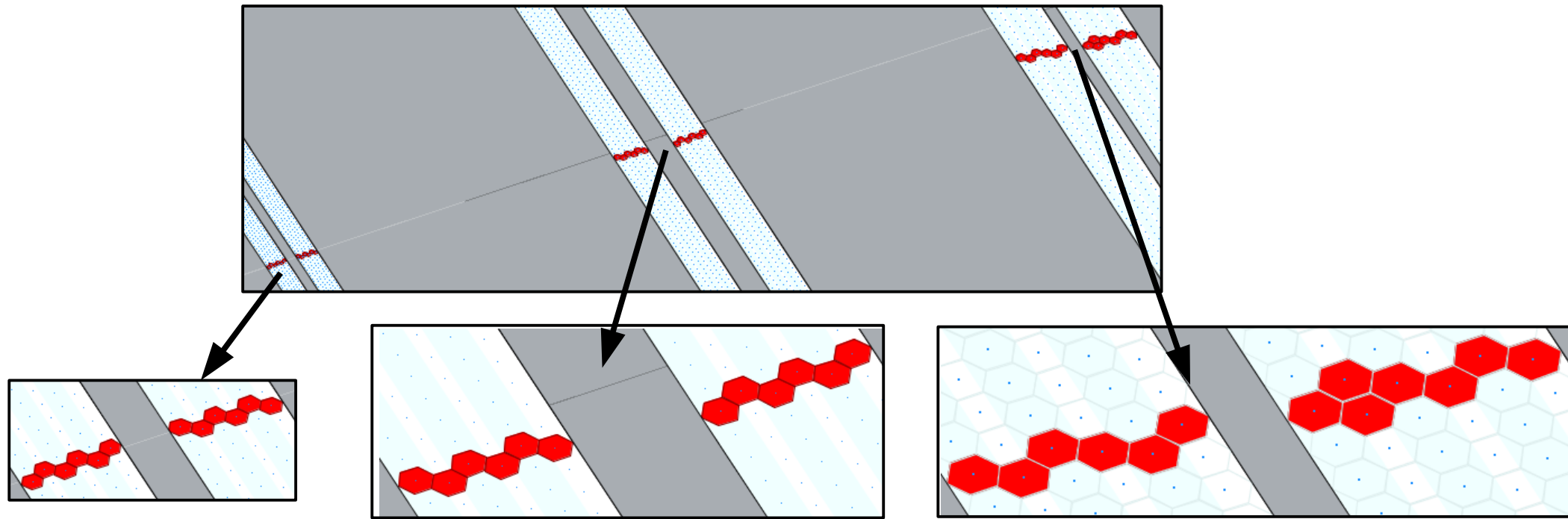
✓ master ⏻ baltzell	EngineProcessor: switch CND engine versions to r	🟢 #688 passed 🔗 283af82 ↗	🕒 7 min 29 sec 📅 13 days ago
✓ master ⏻ Andrey	Merge pull request #75 from JeffersonLab/RICHva	🟢 #687 passed 🔗 930f1fc ↗	🕒 7 min 14 sec 📅 13 days ago
✓ RICHvalidation ⏻ Andrey Kim	set default java version to 8 in maven for rich	🟢 #685 passed 🔗 9fcc5aa ↗	🕒 6 min 42 sec 📅 14 days ago
↓ RICHvalidation ⏻ Andrey Kim	added RICH to reconstrcution.yaml	🔴 #684 errored 🔗 75a1e62 ↗	🕒 1 min 47 sec 📅 14 days ago
✓ master ⏻ Andrey Kim	added preliminary RICH reconstruction engine	🟢 #683 passed 🔗 967e313 ↗	🕒 6 min 53 sec 📅 14 days ago
✓ validation-devel2 ⏻ Nathan Harrison	Build script fix	🟢 #681 passed 🔗 24d0040 ↗	🕒 15 min 53 sec 📅 17 days ago



✓ master baltzell	EngineProcessor: switch CND engine versions to r	🟢 #688 passed 283af82 ↗	🕒 7 min 29 sec 📅 13 days ago
✓ master Andrey	Merge pull req	<b>For every change to the repository:</b> <ul style="list-style-type: none"><li>- Builds the project and checks for errors</li><li>- Runs validation tests</li><li>- Notifies developers of any problems</li></ul>	
✓ RICHvalidation Andrey Kim	set default jav		
↓ RICHvalidation Andrey Kim	added RICH to reconstrcution.yaml	🔴 #684 errored 75a1e62 ↗	🕒 1 min 47 sec 📅 14 days ago
✓ master Andrey Kim	added preliminary RICH reconstruction engine	🟢 #683 passed 967e313 ↗	🕒 6 min 53 sec 📅 14 days ago
✓ validation-devel2 Nathan Harrison	Build script fix	🟢 #681 passed 24d0040 ↗	🕒 15 min 53 sec 📅 17 days ago

# Validation Example: DC Tracking

\* single simulated electron event with  $p=2.5$  GeV,  $\theta=25$  deg,  $\phi=0$ , torus=-1, solenoid=0



Test checks the following for HBT and TBT:

- has tracking bank
- tracking bank has 1 row
- the one track has charge -1
- $p_x$ ,  $p_y$ ,  $p_z$  are close to the true values

\* These tests are run automatically by Travis CI after every build; returns a failure if tests don't pass

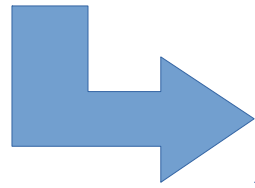


- Recent improvements and updates
  - Shorter and simpler build procedure
  - Improved implementation of unit testing
  - Automatic code coverage reports
  - Automatic SpotBugs reports





JeffersonLab / clas12-offline-software



validation-devel2 branch

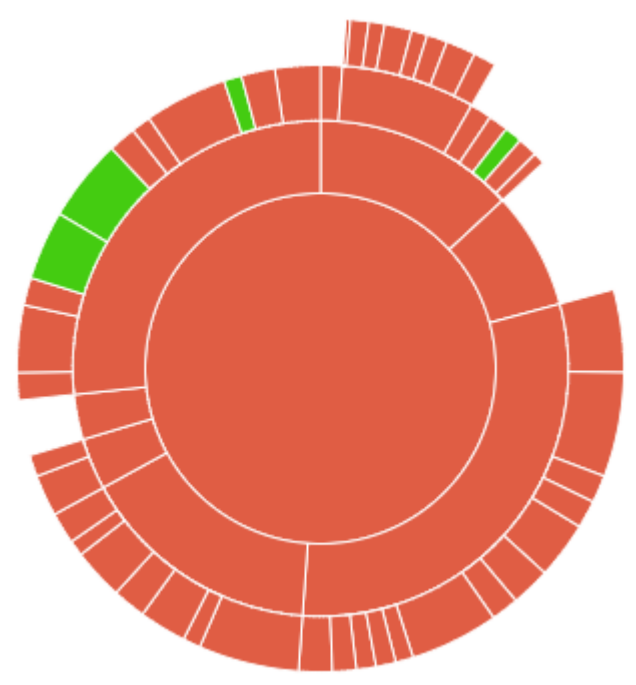
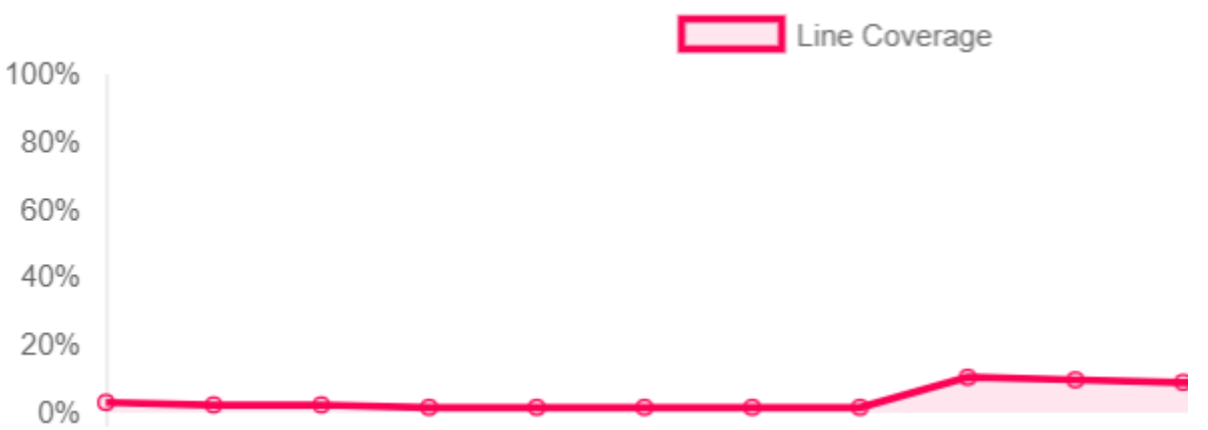
- Cleaner, more efficient build script: 118 lines --> 48 lines
- Major improvement in unit test organization
  - A single command (mvn install) builds the project and runs the unit tests
  - Considerably easier/faster for developers to add new tests, e.g.
    - common-tools/clas-physics/src/main/java/org/jlab/clas/physics/LorentzVector.java
    - common-tools/clas-physics/src/test/java/org/jlab/clas/physics/LorentzVectorTest.java

^^^ note parallel directory structure
  - Automatic code coverage reports from CodeCov

clas12-offline-software build passing codecov 0%

New badge/link to codecov.io





src / main / java / org / jlab / clas

Files	Files	Green	Yellow	Red	Coverage
pdg	133	65	1	67	48.87%
physics	851	191	6	654	22.44%
reactions	175	0	0	175	0.00%
<b>Folder Totals</b> (3 files)	<b>1,159</b>	<b>256</b>	<b>7</b>	<b>896</b>	<b>22.09%</b>
<b>Project Totals</b> (653 files)	<b>53,7...</b>	<b>315</b>	<b>8</b>	<b>53,3...</b>	<b>0.59%</b>



77.46%

## LorentzVector.java

```
51 public void rotateX(double angle) {
52     vector.rotateX(angle);
53 }
54
55 public void rotateY(double angle) {
56     1 vector.rotateY(angle);
57     1 }
58
59 public void rotateZ(double angle) {
60     1 vector.rotateZ(angle);
61     1 }
```

## LorentzVectorTest.java

```
62 @Test
63 public void testRotations() {
64     LorentzVector v =
65         new LorentzVector(0.0, 6.0, 0.0,
66             Math.sqrt(6.0*6.0 + 0.938*0.938));
67
68     v.rotateY(0.1234);
69     assertEquals(v.px(), 0.0, 1e-6);
70     assertEquals(v.py(), 6.0, 1e-6);
71     assertEquals(v.pz(), 0.0, 1e-6);
72
73     v.rotateZ(Math.toRadians(90.0));
74     assertEquals(v.px(), -6.0, 1e-6);
75     assertEquals(v.py(), 0.0, 1e-6);
76     assertEquals(v.pz(), 0.0, 1e-6);
77 }
```

(Test Driven Development (TDD) is a popular development process in which unit tests are written *before* the actual code!)



# SpotBugs

- Runs as a Maven plugin; analyzes the entire code base with a single command (done automatically by Travis CI):
  - `mvn com.github.spotbugs:spotbugs-maven-plugin:spotbugs` (gives warnings)
  - or
  - `mvn com.github.spotbugs:spotbugs-maven-plugin:check` (gives errors)
- Checks for over 400 common bug patterns (which might be missed by the compiler and unit tests)
- `spotbugs-exclude.xml` file provides customization – e.g. ignore non-serious bugs (such as a possible “bug” if `sector < 0`)
- Generates a bug report for each module, e.g.
  - `reconstruction/dc/target/spotbugsXml.xml`
  - `common-tools/clas-geometry/target/spotbugsXml.xml`
- GUI tool for debugging



## SpotBugs - Example

**Eq: equals method overrides equals in superclass and may not be symmetric (EQ\_OVERRIDING\_EQUALS\_NOT\_SYMMETRIC)**

This class defines an equals method that overrides an equals method in a superclass. Both equals methods use `instanceof` in the determination of whether two objects are equal. This is fraught with peril, since it is important that the equals method is symmetrical (in other words, `a.equals(b) == b.equals(a)`). If B is a subtype of A, and A's equals method checks that the argument is an instanceof A, and B's equals method checks that the argument is an instanceof B, it is quite likely that the equivalence relation defined by these methods is not symmetric.

# Summary/Conclusions

- The CLAS12 software team is working to create a culture in which our code is of a high quality, clean, easy to use, and reliable
- Over the past ~1 year, GitHub + Travis CI has been a very successful code management and validation system
- The improvements described here will soon be merged into the master branch and are expected to build upon an already strong foundation