

# AIDA Proxy Unit Tests

---

Hurung-Chun Lee

Academia Sinica Computing Centre,  
Taiwan



# AIDA Proxy Layer

## ❖ C++ proxy classes to AIDA interfaces

- ❑ “Value semantics” for AIDA objects
- ❑ Implemented using the “Proxy” pattern, very easy !
- ❑ Based only on AIDA Interfaces
  - ➔ *no dependency on a given implementation*

## ❖ Initially “hiding” of AIDA object management

- ❑ AIDA tree is not exposed to users but hid in the Proxy implementation

## ❖ Keeping the functionality and signatures of AIDA

- ❑ “re-shuffling” of factory methods to object constructors



# Purposes of the Tests

- ❖ **Checking the functionalities of AIDA Proxy**
- ❖ **Checking the consistencies between different histogram implementations through AIDA Proxy**
  - Or obtaining the differences between implementations
- ❖ **Creating the unit test framework for AIDA Proxy on which the new tests can be easily adapted**



# Current Status

## ❖ **pi\_aida Classes covered by the tests**

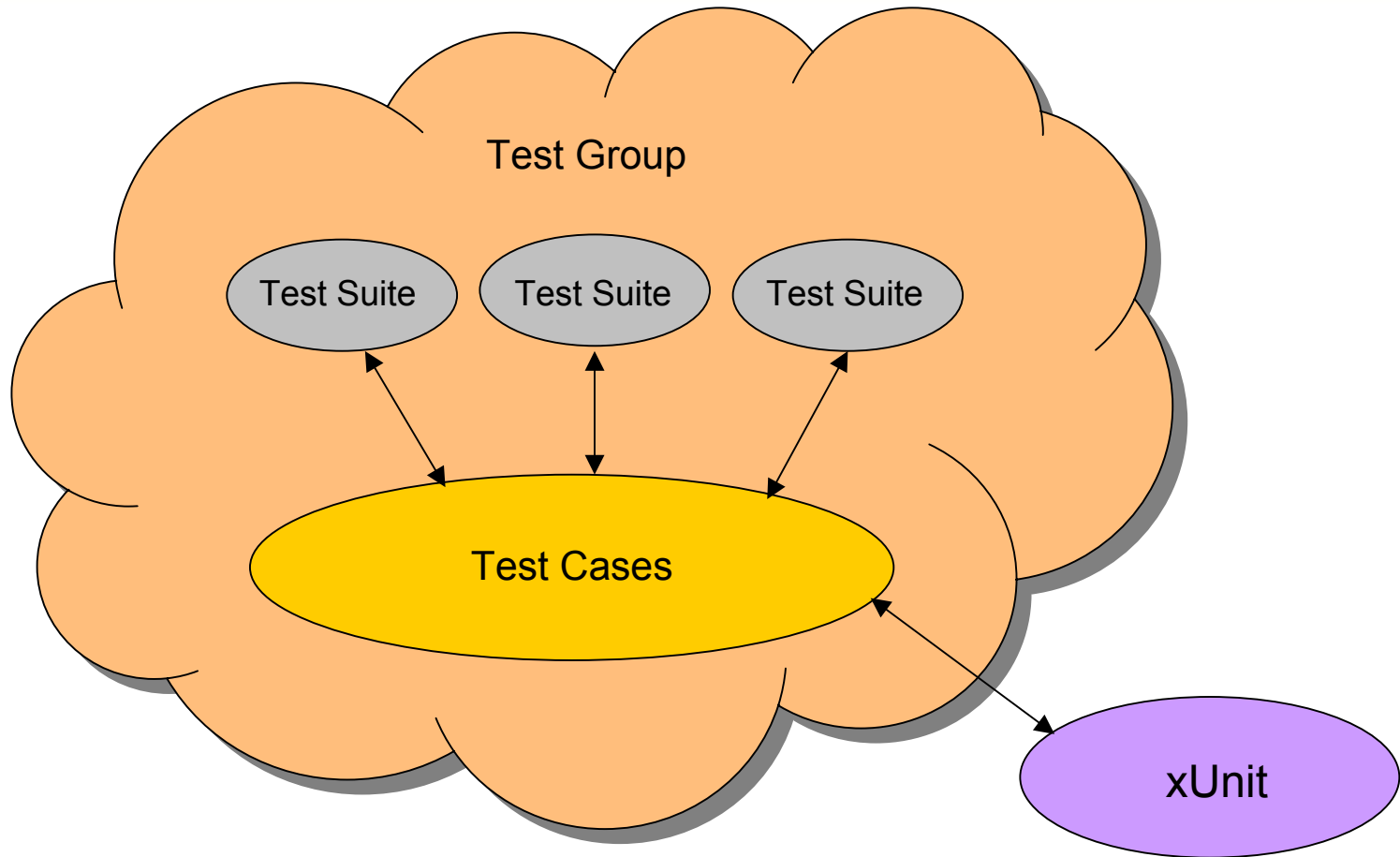
- ❑ Histogram1D, 2D, 3D
- ❑ Cloud1D, 2D, 3D
- ❑ Profile1D, 2D
- ❑ DataPointSet
- ❑ HistoProjector, ProxyStore

## ❖ **Over 1000 CppUnit assertions for consistency tests between Native and ROOT implementations**

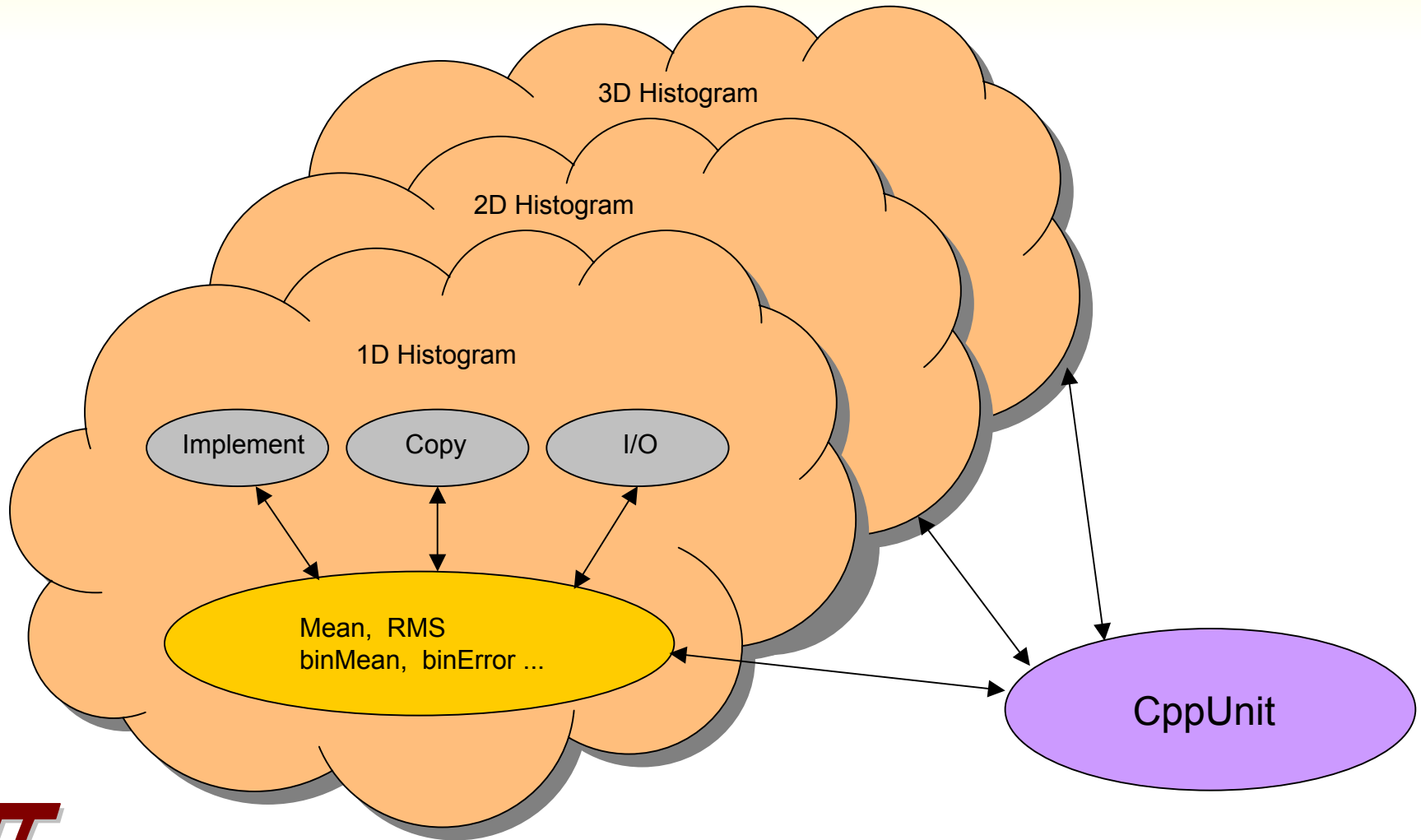
## ❖ **All independent tests have been integrated in Oval and QMtest**



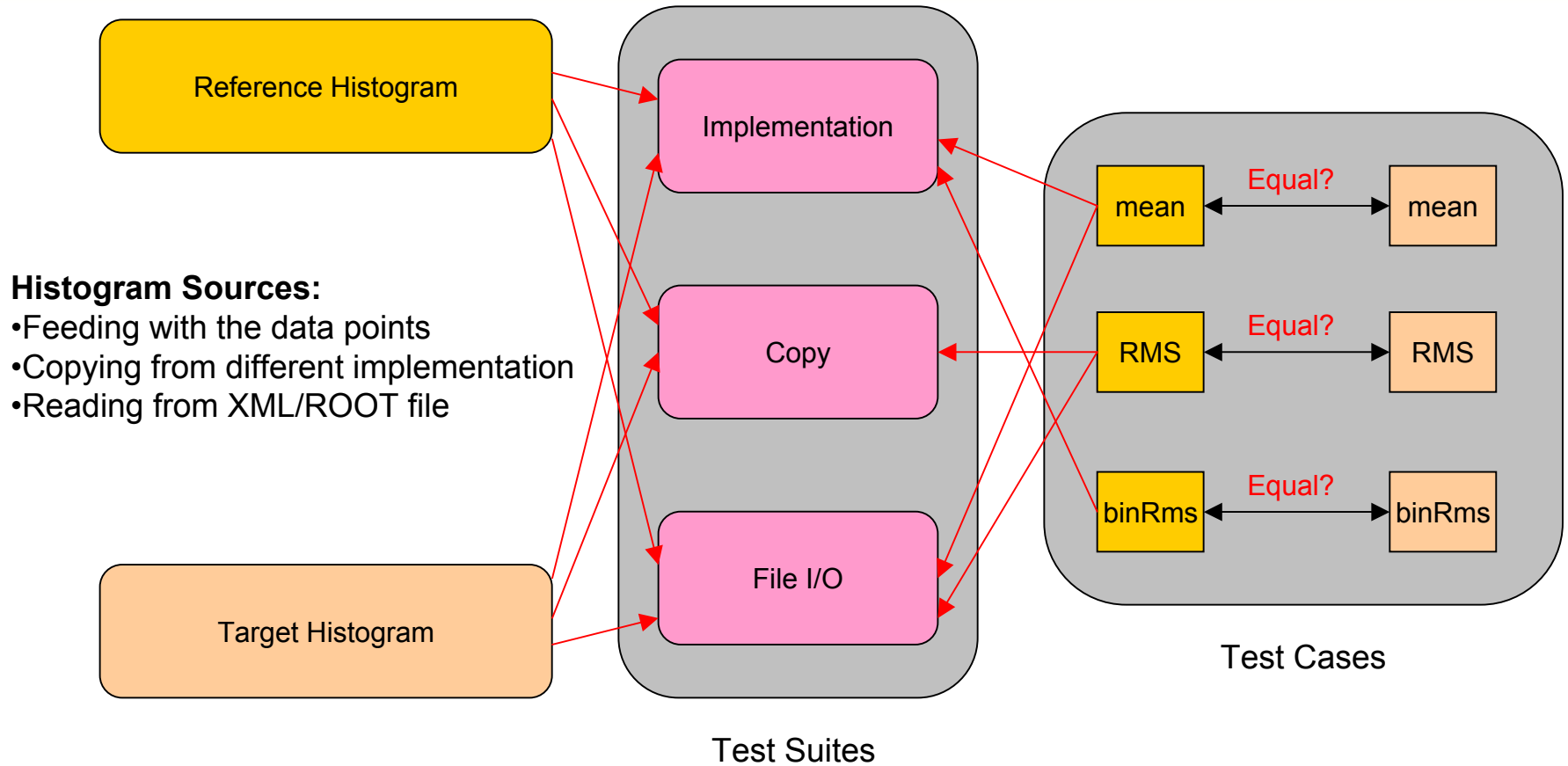
# Hierarchical Structure of Unit Tests



# Unit Tests of AIDA Proxy

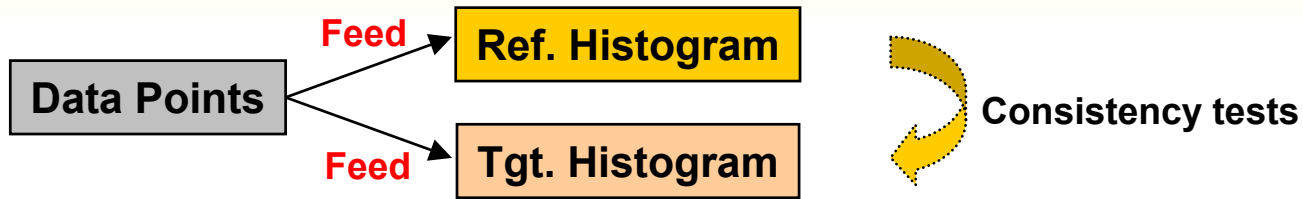


# Testing Logic

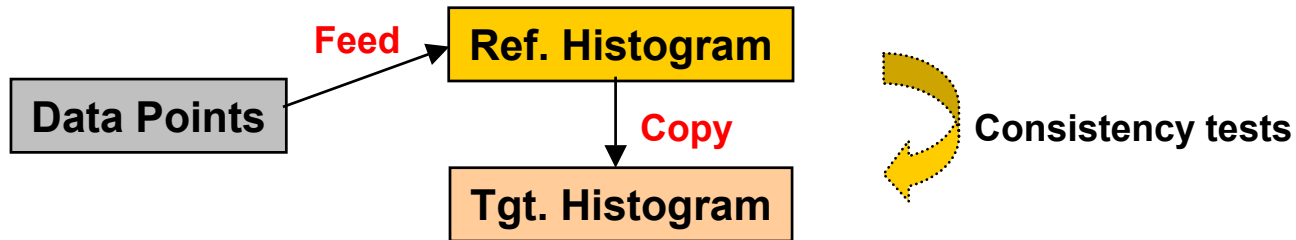


# Test Suite Categories

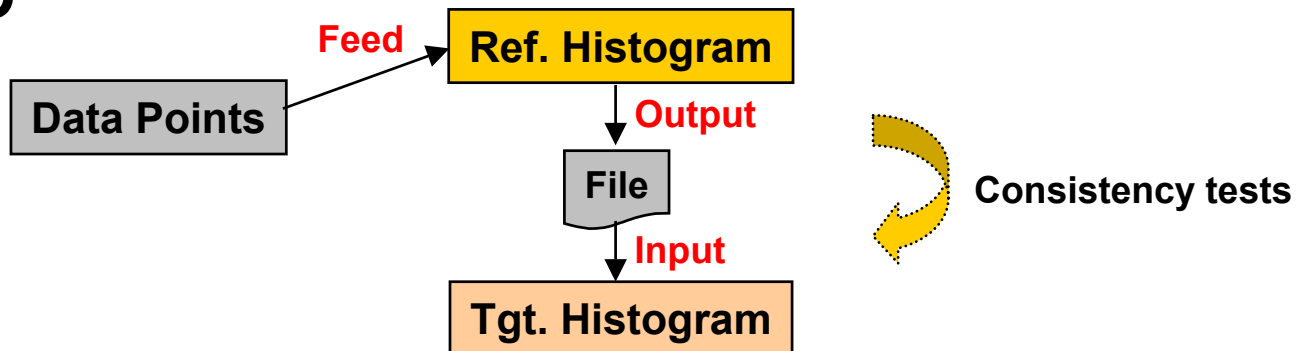
## ❖ Implementation



## ❖ Copy

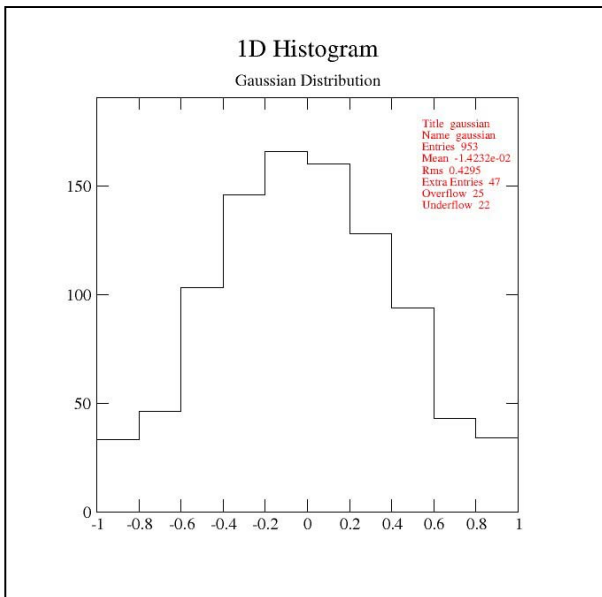
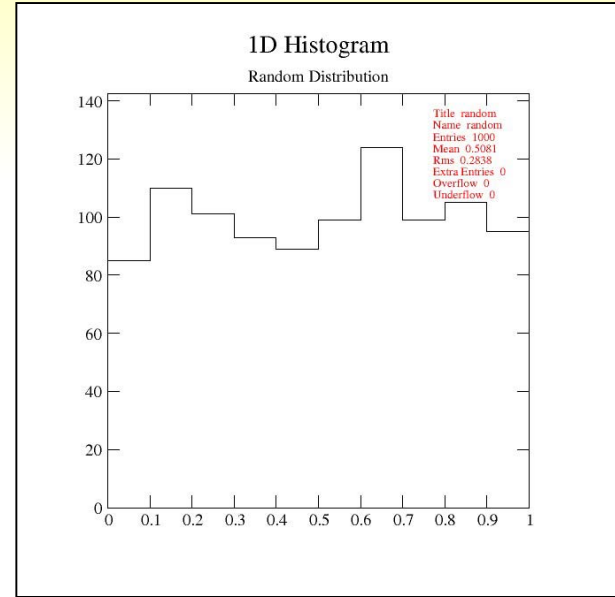
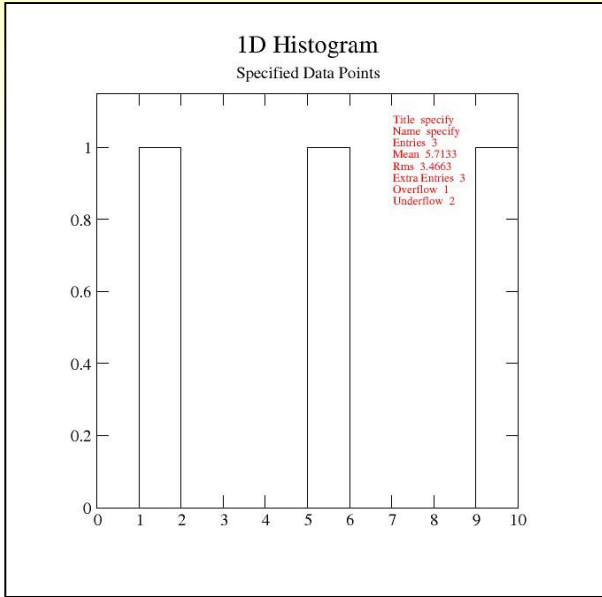


## ❖ File I/O





# Data Points (Histogram View)



- ❖ **Specific numbers**
  - ❑ (-2.99, -1.05, 1.5, 5.65, 9.99, 10.01)
- ❖ **Random distribution**
  - ❑ 1000 numbers
  - ❑ Range: [0,1]
- ❖ **Gaussian distribution**
  - ❑ 1000 numbers
  - ❑ Mean:0 StdDev: 0.5



# Code examples (Test Cases)

## ❖ Test case implementations of CppUnit::TestFixture

```
class Histogram1D_TCase : public CppUnit::TestFixture {
    pi_aida::Histogram1D *refHist, *trgHist;
    void tstMean();
    void tstBinMean();
    void tstTitle();
}
```

Histogram1D\_TCase.h

### ❑ Tests of global statistics

```
void Histogram1D_TCase::tstMean() {
    CPPUNIT_ASSERT_DOUBLES_EQUAL(refHist->mean(),trgHist->mean(),10e-6);
}
```

Histogram1D\_TCase.cpp

### ❑ Tests of local (per bin) statistics

```
void Histogram1D_TCase::tstBinMean() {
    for(unsigned int i=0; i<refHist->axis().bins(); i++) {
        CPPUNIT_ASSERT_DOUBLES_EQUAL(refHist->binMean(i),trgHist->binMean(i),10e-6);
    }
}
```

### ❑ Tests of annotations

```
void Histogram1D_TCase::tstTitle() {
    CPPUNIT_ASSERT_EQUAL(refHist->title(),trgHist->title());
}
```



# Code Example (Test Suites)

## ❖ Ref./Tgt. Histogram and Test Case Specification

```
class Histogram1D_Impl : public Histogram1D_TCase {  
  
    CPPUNIT_TEST_SUITE(Histogram1D_Impl);  
    CPPUNIT_TEST(tstMean);  
    CPPUNIT_TEST(tstRms);  
    CPPUNIT_TEST(tstTitle);  
    CPPUNIT_TEST_SUITE_END();
```

**Test Case Specification**

**Histogram Initialization**

```
refHist = new pi_aida::Histogram1D("1D Native Histogram",10,0,1,  
                                   "AIDA_Native_Histogram");  
trgHist = new pi_aida::Histogram1D("1D ROOT Histogram",10,0,1,  
                                   "AIDA_Root_Histogram");
```

**Data Point Insertion**

```
dg = new DataGen;  
dg->random();  
for(unsigned int i=0; i<dg->feeds.size(); i++){  
    refHist->fill(dg->feeds[i],dg->weight[i]);  
    trgHist->fill(dg->feeds[i],dg->weight[i]);  
}  
}
```



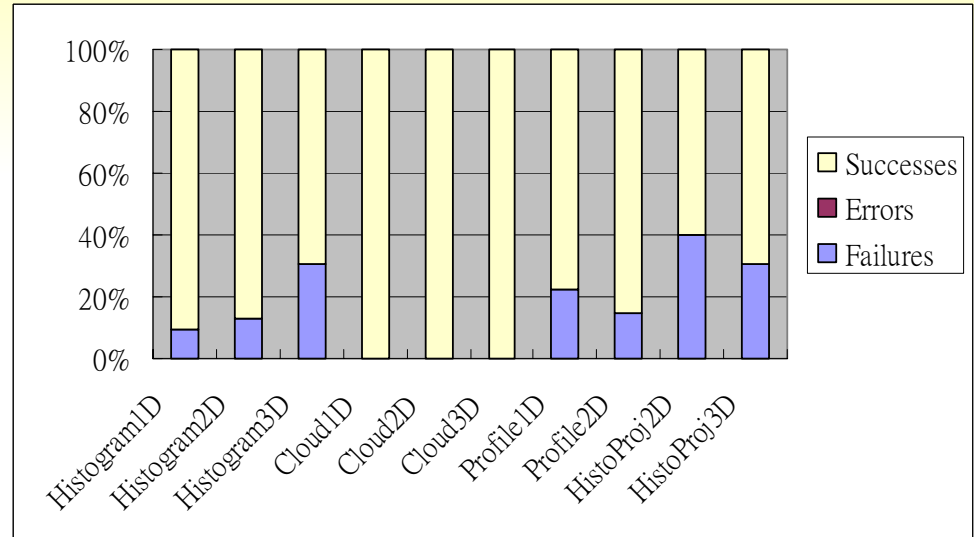
# Testing Results

❖ **PI version: PI\_1\_0\_0**

❖ **161 (~15%) failures in 1051 CppUnit assertions**

❖ **Failures are due to:**

- ❑ The mixture of bugs in AIDA Proxy and implementation differences

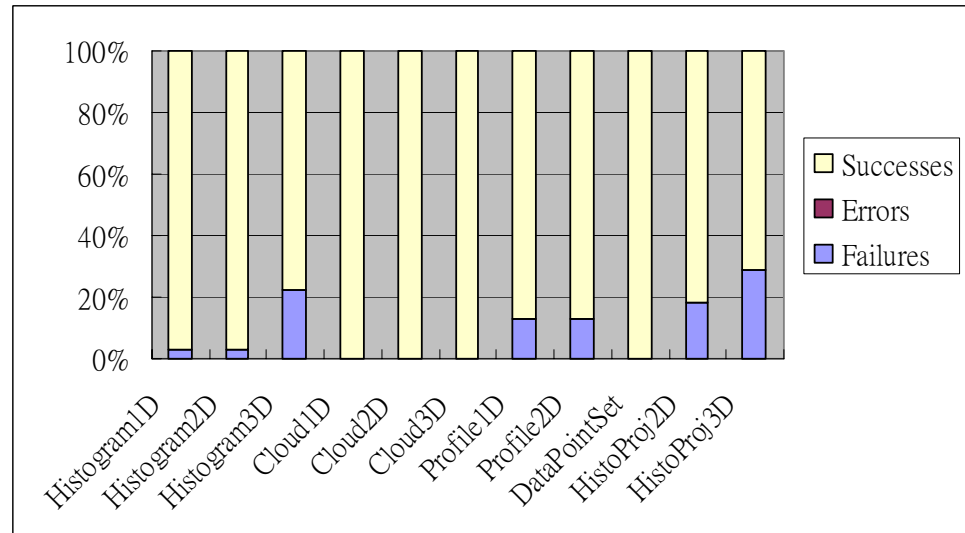


❖ **PI version: PI\_1\_1\_0-pre1**

❖ **104 (~9%) failures in 1164 CppUnit assertions**

❖ **Failures are due to:**

- ❑ Implementation differences
  - Root takes the binCentres instead of the values to calculate the global mean in H3D and Profiles
  - Root doesn't store the binMean
  - Error treatment in Profile



# Summary & Future works

## ❖ Summary

- ❑ New CppUnit based test package for AIDA Proxy Layer has been available from PI release 1.0.0 (*Thanks Lorenzo, Andreas and Vincenzo for the inputs and discussions*)
- ❑ New tests can be easily added on by extending one of the three (Group/Suite/Case) levels in the hierarchical testing structure
- ❑ All independent tests were integrated in Oval and QMtest (*Thanks Manuel*)

## ❖ Future Works

- ❑ Unit tests on Tuple, Fitter and Minimizer will be included in the coming PI release

