

# J A I D A

*(FreeHEP AIDA)*



- Java implementation of AIDA
- Based on FreeHEP toolkit
- Usable from Java, C++, Python, P<sub>N</sub>uts
- Accessible via JAS3 GUI
- Supports Ascii, XML, H<sub>B</sub>ook, Root, SQL<sup>(LAL)</sup>, ... storage
- Available as a JSP Web Service
- Hosted at SLAC
- Illegal in LCG/CERN

<http://java.freehep.org/jaida>  
<http://jas.freehep.org/jas3>  
<http://java.freehep.org/aidajni>  
<http://java.freehep.org>  
<http://aidatld.freehep.org>

# Distributed Data

```
// Prepare Factories
IAnalysisFactory af = IAnalysisFactory.create();
ITreeFactory trf = af.createTreeFactory();

// Open local Tree
ITree tree1 = trf.create("localFile.aida", "xml");

// Open remote Root Tree
ITree tree2 = trf.create("root://rootsrv.some.where/remoteFile.root", "root");

// Open remote SQL Tree
ITree tree3 = trf.create("jdbc:mysql://mysqlsrv.some.place/Database", "sql");

// Find NTuple in remote SQL Tree
ITuple inTuple = tree3.find("myInputTuple");

// Prepare factory for output NTuple
ITupleFactory tf = af.createTupleFactory(tree1);

// Define cut
IFilter filter = tf.createFilter("Pt > 6.0");

// Apply cut on remote (SQL) Ntuple and write resulting Ntuple into local file
ITuple outTuple = tf.createFiltered("myOutputTuple", inTuple, filter);
```

Root analysis objects  
are interpreted as  
AIDA objects  
(e.g. TH1 -> IHistogram1D)

SQL Works with  
LCGPool AttributeLists.

Cut/Query is performed  
inside remote SQL database.  
Results are available locally.

*Data and intensive processing distributed, Client local.*

# JAS3 (Java Analysis Studio)



- GUI
- Dynamic class re-loading
- Grid integration
- Easily extensible via plugins (many exist)

- AIDA objects
- Programs
- Cuts

- Program outputs
- Compiler outputs
- Python command-line
- PNuts command-line

The screenshot shows the JAS3 interface with the following components:

- File Explorer:** A tree view on the left showing a project structure with folders like 'tree-0', 'Programs', and 'Histogram 1D'.
- Code Editor:** The central window displays Java code for 'Fit.java', including imports for 'hep.aida.\*' and 'java.util.Random', and a main method that creates a histogram and fills it with random data.
- Data Table:** A window titled 'tuple' shows a table with columns 'iFlat', 'fGauss', and 'fFlat', containing numerical data points.
- Plots:** Three plots are visible: a 1D histogram (h1dG) with a Gaussian fit, a 2D scatter plot (h2d), and a 1D histogram (h1dF) with a Gaussian fit.
- Output Console:** At the bottom, it shows 'Mimuit Chi2=0.9074226907136734' and 'Uncmin Chi2=0.9074224707742339'.

Web Browser

- Java source
- Python source
- PNuts source

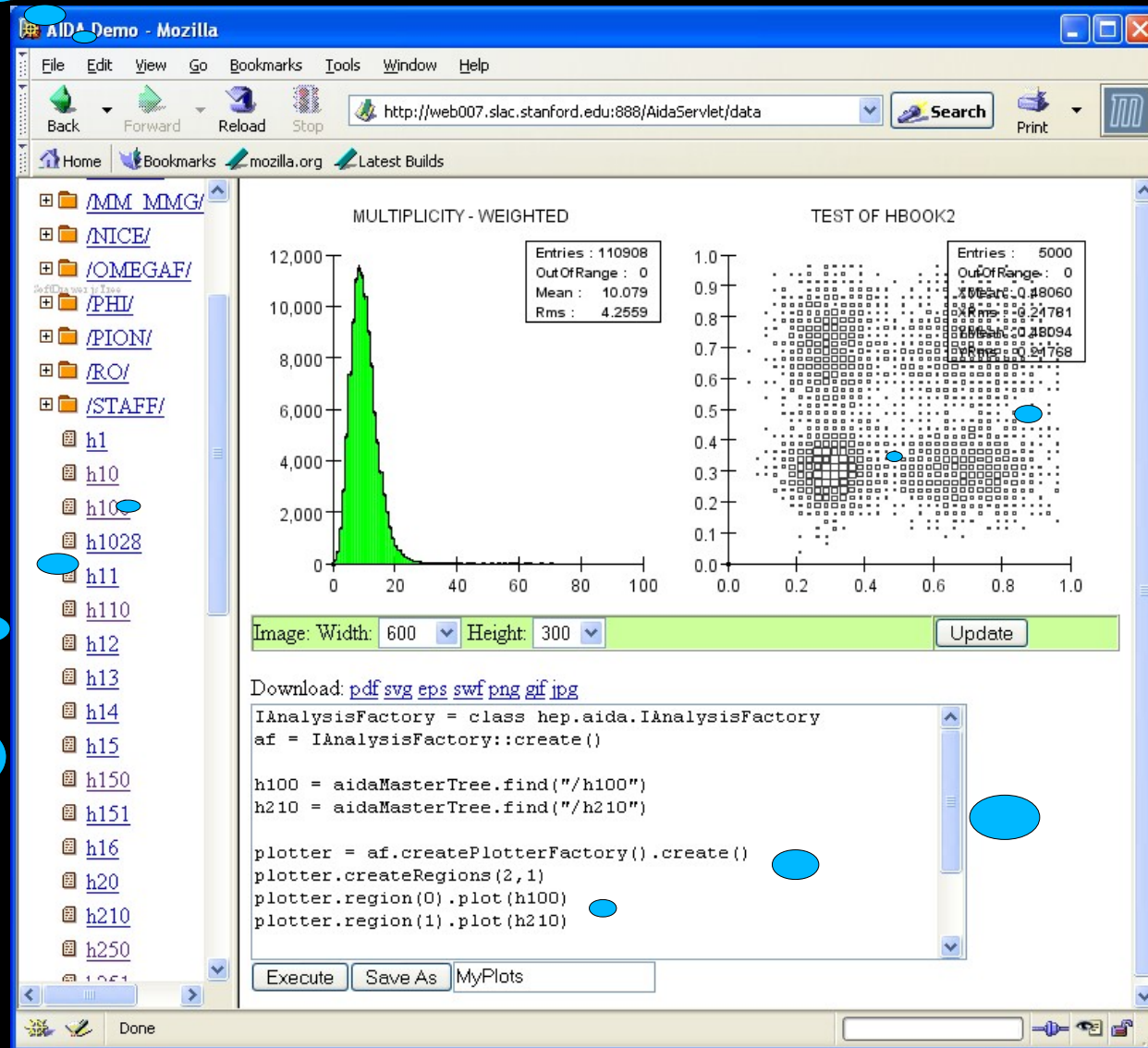
Table view

Graphical output

Web Browser

# Web Service (JSP)

FREE



Results

Data  
on the Server

Scripting,  
runs on the Server

- Can be used together with other Web Services – distributed Analysis.
- Can be used directly from Client code (i.e. without Web Browser).

# SQLTuple

- *SQL table accessed as JAIDA ITuple*
- *Compatible with*
  - *Other AIDA implementations*
  - *LCG Pool AttributeList*
- *No SQL in the code, read from text file*
- *Available also as JAS3 plugin and Web Service*
- *Any DB with JDBC driver can be supported*
  - *MySQL, PostgreSQL, Oracle, McKoi included; Cloudscape (now Apache Derby) and HSQLDB tested*
- *Used in Atlas DC2*
- *Developed in LAL*

