

javaROOT

A Java Interface to ROOT

M. Lynch M. Stanitzki J. Strube

Rutherford Appleton Laboratory
STFC

August 2009

Outline

- 1 Introduction
 - javaROOT
 - Prerequisites
 - Building javaROOT
- 2 Implementation
 - SWIG
 - Library structure
- 3 Features
 - Histograms
 - Trees
 - Example

javaROOT

What is javaROOT?

- Allows creation of (not reading/modification of existing) ROOT files in Java.
- Provides access to some of the most commonly used features.
- (Further processing can then be carried out in dedicated ROOT scripts.)
- Runs on both Linux and Windows (tested on Scientific Linux and Windows XP).
- Eclipse integration—auto-completion works out of the box.
- Collection of self-contained examples and tests included with source.

“Lets you get the data into ROOT.”

Prerequisites

Tested on Linux and Windows. Code is platform independent.

- Core tools:
 - ROOT, Java JDK, SWIG, Doxygen (for documentation)
- Linux
 - GCC
- Windows
 - Microsoft Windows SDK, MinGW (for makefile)

Building javaROOT on Linux

- Setup ROOT environment.
- Run makefile.
- Use libraries. (Slight care needed with `$LD_LIBRARY_PATH`.)

Building javaROOT on Windows

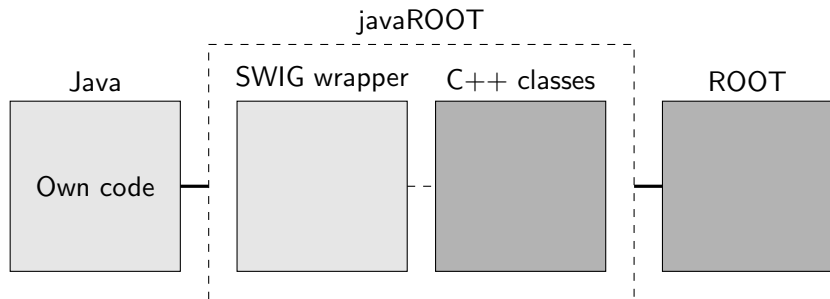
- Set environment variables: %ROOTSYS% and %JDKHOME%.
- Run makefile.
- Use libraries. (Care with locations of DLL's necessary).

SWIG

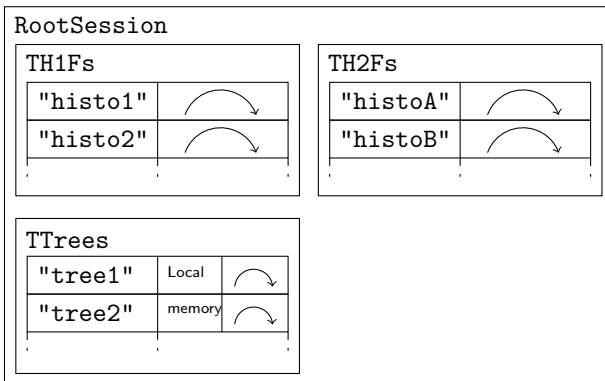
Simplified Wrapper and Interface Generator (<http://www.swig.org/>)

- Development tool to connect C++ programs with another high-level language—in our case, Java.
- The “magic behind the scenes”—lets us join up Java and ROOT without having to resort to manual JNI.
- Specify an interface file, and the C++ headers, and Java classes corresponding to our original code are automatically generated. Calls to the Java code are automatically mapped to their parent functions on the C++ side.

Library structure



C++ classes



Histograms

- Fill TH1Fs, TH1Ds, TH2Fs, TH2Ds and TProfiles.
- Syntax very close to corresponding ROOT calls, e.g.:

```
RootSession sess = new RootSession( "FromJava.  
    root", "RECREATE", "Test", 1 );  
sess.newTH1F( "demoA", "Title1", 100, 0, 1 );  
sess.fillTH1F( "demoA", 0.9 );  
sess.delete();
```

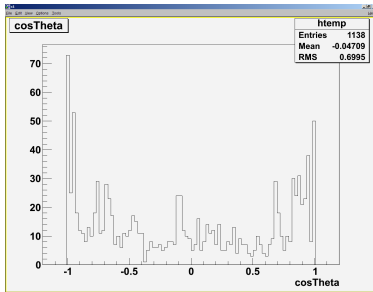
Trees

- Fill TTrees.
- Basic structure—only simple branches.
- Only supports branches of floats, ints, doubles, bools, TVector3s and TLorentzVectors. (Additional types could be added manually.)
- For convenience, function calls on Java side use the freehep classes BasicHep3Vector and BasicHepLorentzVector which are mapped to the native ROOT types when stored in the tree (see bundled examples).

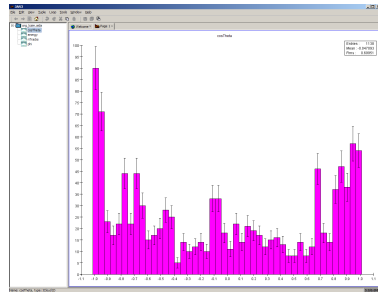
Trees

```
RootSession sess = new RootSession( "FromJava.root"  
    , "RECREATE", "Test", 1 );  
sess.newTTree( "demoT", "TestTree", 99 );  
sess.branchTTreeFloat( "demoT", "fX" );  
sess.branchTTreeFloat( "demoT", "fY" );  
5 sess.branchTTreeFloat( "demoT", "fZ" );  
sess.setupTTree( "demoT" );  
sess.fillBranchFloat( "demoT", "fZ", (float) r.  
    nextGaussian() );  
sess.fillBranchFloat( "demoT", "fX", (float) r.  
    nextGaussian() );  
sess.fillBranchFloat( "demoT", "fY", (float) r.  
    nextGaussian() );  
10 sess.fillTTree( "demoT" );  
sess.delete();
```

Example



(a) ROOT



(b) JAS3

Summary

- Provides a **Java interface** to some of the most commonly used features.
- **Syntax** very similar to ROOT's so easy to learn.
- Future work
 - Add support for more features as needed?
 - Create installer, registering libraries.
 - Mac support.
- Available to download:

svn checkout

<https://heplnm060.pp.rl.ac.uk/repos/javaROOT/tags/0.1.0>