javaROOT

How to get your data from java into ROOT

Marcel Stanitzki
Jan Strube
Rutherford Appleton Laboratory

Premise

- org.lcsim is a simulation and reconstruction framework
  - "Do one job and do it well"
- Analysis goes on at higher level
  - Histogramming and Plotting
  - Likelihood Fits
  - Statistical Tools
  - Multivariate Classifiers
- HEP is growing a monoculture
  - LHC Analysis expertise is only made available through ROOT
  - Data input into ROOT pretty much only in ROOT files

javaROOT

- Keep it simple
  - Minimal maintenance → only minimal functionality
  - Data out of org.lcsim, into ROOT
  - Only export a few constructs
- Use case: Most people want to write out
  - Histograms
  - ntuples (TTrees)
  - javaROOT does exactly that (and only that)
- Can't keep up with ROOT “development”
  - Use automatic binding generator
  - Does not depend on a specific ROOT version
- Works on Linux and Windows
  - Run on your laptop
- No iphone / Android release planned

How to Obtain and Build (on Linux)

1. Setup the ROOT environment: source <wherever you installed ROOT>/bin/thisroot.sh
2. Setup the JAVA_HOME environment variable. The java executable is supposed to be found in $JAVA_HOME/bin/java

```
3. svn co https://heplnm060.pp.rl.ac.uk/repos/javaROOT/trunk
javaROOT
4. cd javaROOT
5. make OSTYPE=linux-gnu
```

Code Example

create histograms:

```
sess.newTH1F( "demoA", "demoA", 100, 0, 1 );
for( int i = 0; i < 4096; i++ ) {
    sess.fillTH1F( "demoA", (float) r.nextGaussian() );
}
sess.delete();
```

create TTrees:

```
sess.newTTree( "demoT", "TestTree", 99 );
sess.branchTTreeFloat( "demoT", "fX" );
for( int i = 0; i < 4096; i++ ) {
    sess.fillBranchFloat( "demoT", "fX", (float) r.nextGaussian() );
    sess.fillTTree( "demoT" );
}
sess.delete();
```

Constructs

- Histograms
  - TH1F, TH1D
  - TH2F, TH2D
  - TProfile
- TTree with branches of type
  - float, int, double, bool
  - TVector3, TLorentzVector
  - std::vector<float>, std::vector<double>
  - std::vector<int>, std::vector<bool>
  - std::vector<TVector3>
  - std::vector<TLorentzVector>
- Should cover all of your use cases. Extensions may be considered

Help Is Available

javaROOT comes with
Instructions on SLAC confluence
- API documentation (doxygen)
- Examples
- Test cases

Maintained on best effort basis. If it breaks, please turn in both pieces to
- Marcel Stanitzki
- Jan Strube

for inspection and replacement.