org.lcsim
Reconstruction and Analysis framework for ILC Detectors

Tony Johnson
SLAC
July 2006

org.lcsim: Contents

- Overview/Goals
- Geometry/Conditions/Detector system
- Reconstruction overview/status
- Using org.lcsim with JAS3
- Using org.lcsim with WIRED4
- Becoming an org.lcsim Developer
- Where next?

org.lcsim Goals

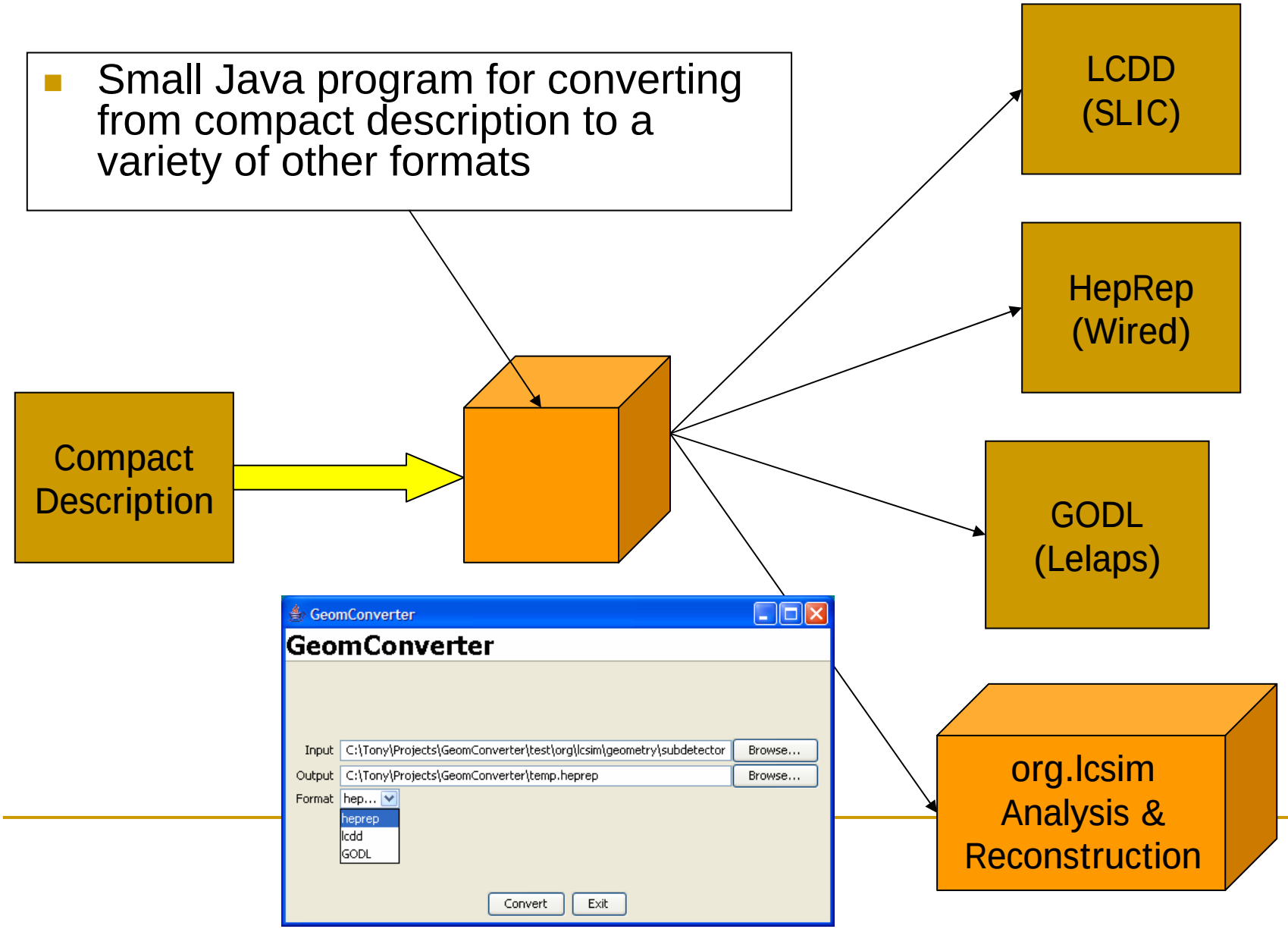
- “Second generation” ILC reconstruction/analysis framework
  - Builds on hep.lcd framework used since 1999
  - Full suite of reconstruction and analysis tools
- Uses LCIO for IO and as basis for simulation, raw data and reconstruction event formats
  - Isolate users from raw LCIO structures
  - Maintain full interoperability with other LCIO based packages
- Detector Independence
  - Make package independent of detector, geometry assumptions so can work with any detector
  - Read properties of detectors at runtime
- Written using Java (1.5)
  - High-performance but simple, easy to learn, OO language
  - Enables us last 10 years of software developments in the “real world”
- Ability to run standalone (command line or batch) or in JAS3 or IDE such as Netbeans, Eclipse

org.lcsim: Compact Geometry Description

- org.lcsim uses “Comact Geometry Description” to define detector
  - Simple XML format for describing ILC detectors
  - Handles typical ILC detector geometries
    - Range of detectors handled is extensible (by writing Java modules)
- Allows rapid prototyping of new detector geometries
- Does not require network access or installation of database software to run
- Automatic generation of full Geant4 LCDD geometry for full compatibility with SLIC

org.lcsim: Geometry Converter

- Small Java program for converting from compact description to a variety of other formats

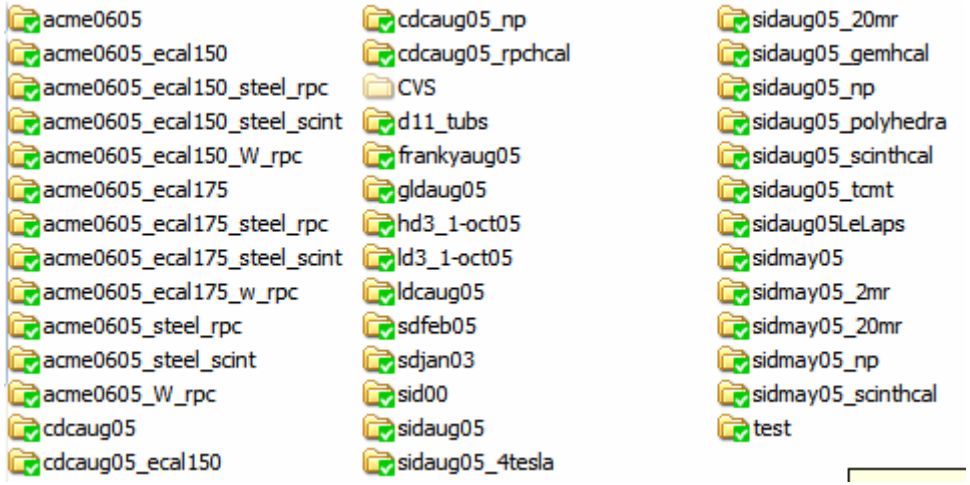


org.lcsim Conditions Data

- Provide access to a extensible set of conditions for each detector including:
  - Detector Geometry
  - Algorithm Specific Constants
    - E.g. FastMC smearing parameters
- Doesn't make assumptions about format of data
- Doesn't rely on internet access, or local database installation
- Detector Constants stored in .zip file
  - Typically contains:
    - Compact geometry file
    - Set of (ascii) constants for standard algorithms
  - Can additionally contain:
    - Arbitrary files (xml, ascii, binary) needed by other algorithms
    - Other geometry formats (HepRep, LCDD)
    - Full fieldmap
- To define a new detector just create a new .zip file.

Available Detector Descriptions

- Although detector descriptions can live anywhere we maintain a CVS repository of detector descriptions
  - Exported to org.lcsim web site for automatic download
- 40 detector variants as of July 2006
- Many SiD variants, but also some gld, ldc



- You are welcome to contribute more

Org.lcsim Reconstruction

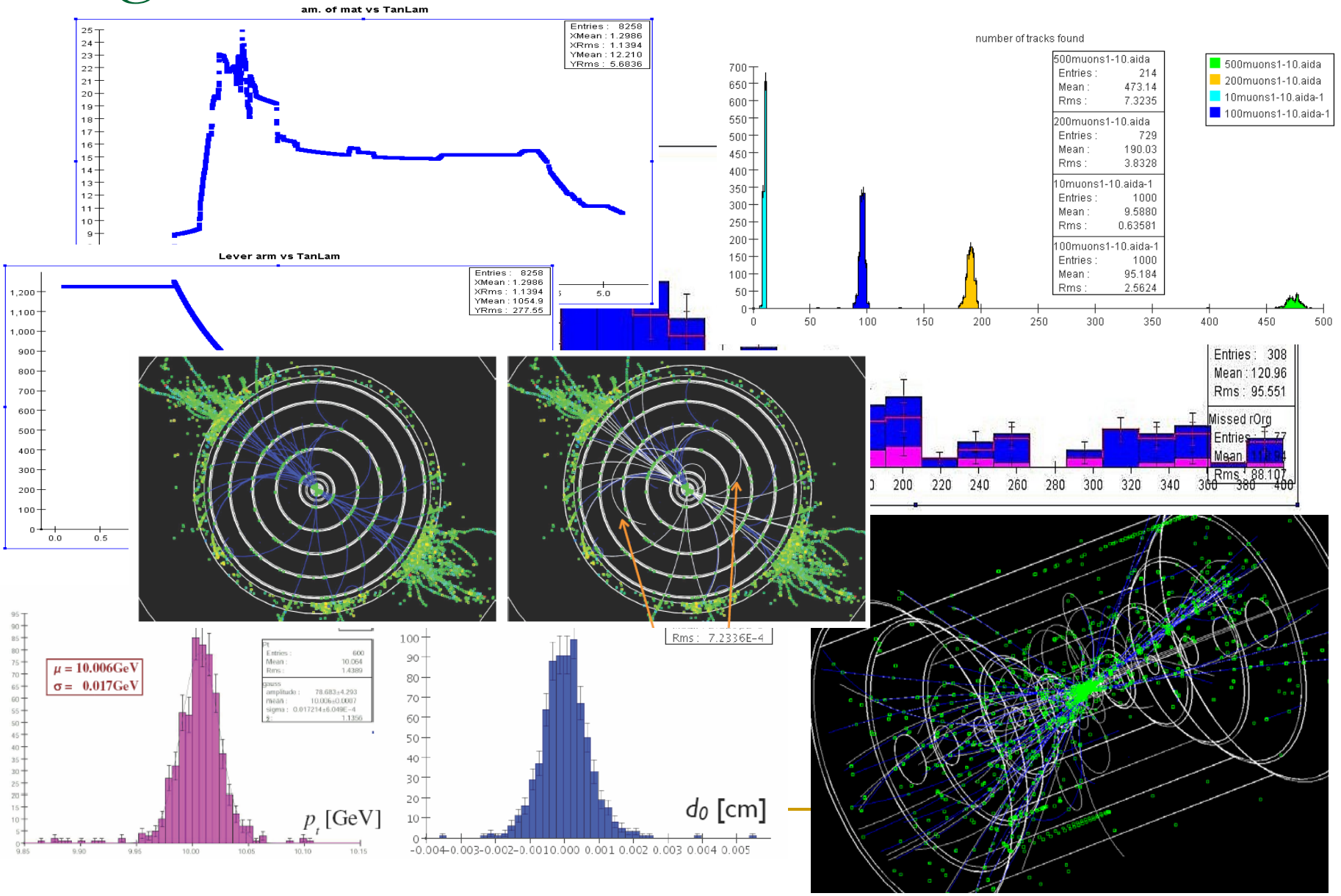
- Reconstruction package includes:
  - Physics utilities:
    - Jet finders, event shape routines
    - Diagnostic event generator, stdhep reader/translator
    - Histogramming/Fitting/Plotting (AIDA based)
    - Event Display
    - Processor/Driver infrastructure
  - Fast MC
    - Track/Cluster smearing
  - Reconstruction
    - Cheaters (perfect reconstruction)
    - Detector Response
      - CCDSim, Digisim
    - Clustering Algorithms
      - Cheater, DirectedTree, NearestNeighbour, Cone
    - Tracking Finding/Fitting Algorithms
      - TRF,
    - Muon Finding, Swiming
    - Vertex Finding (ZvTop)

org.lcsim: Contrib Area

- Goal of org.lcsim is not to provide “A single reconstruction package” but rather a framework into which reconstruction algorithms can be plugged.
- We encourage users to contribute code to the “contrib” area as soon as possible.
  - Important to encourage collaboration, reuse, and as learning tool.
- Many contributions added in last year:
  - HMatrix cluster analysis
  - VertexFitter
  - PFA algorithms/template
  - SODTracker
  - Garfield Tracker
  - Calorimeter Cell Ganging
  - FastMC improvements
  - Tracking finding/fitting
  - MIP Finder
  - Minimum Spanning Tree Clustering

org.lcsim results

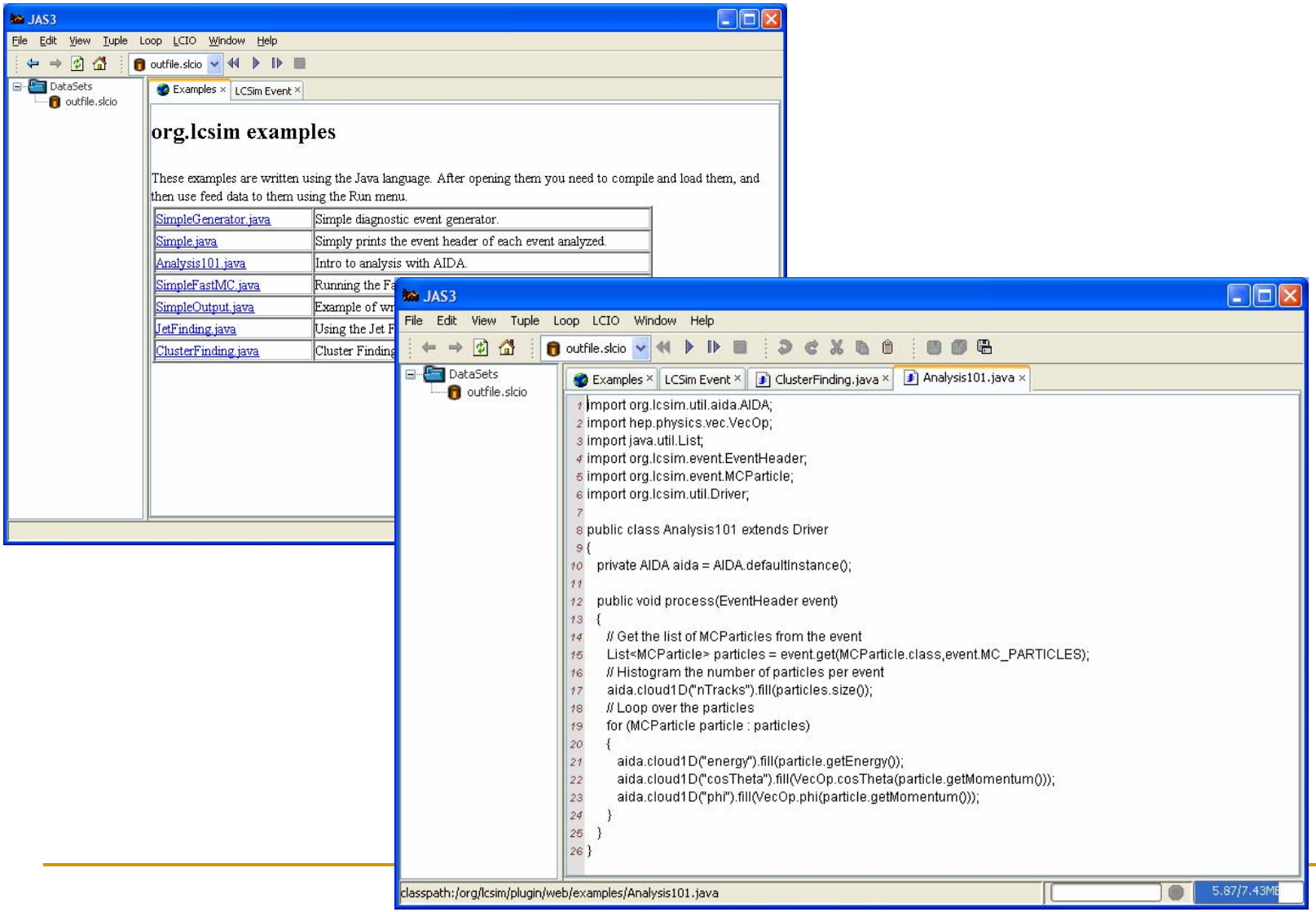
(See many other talks at this workshop)



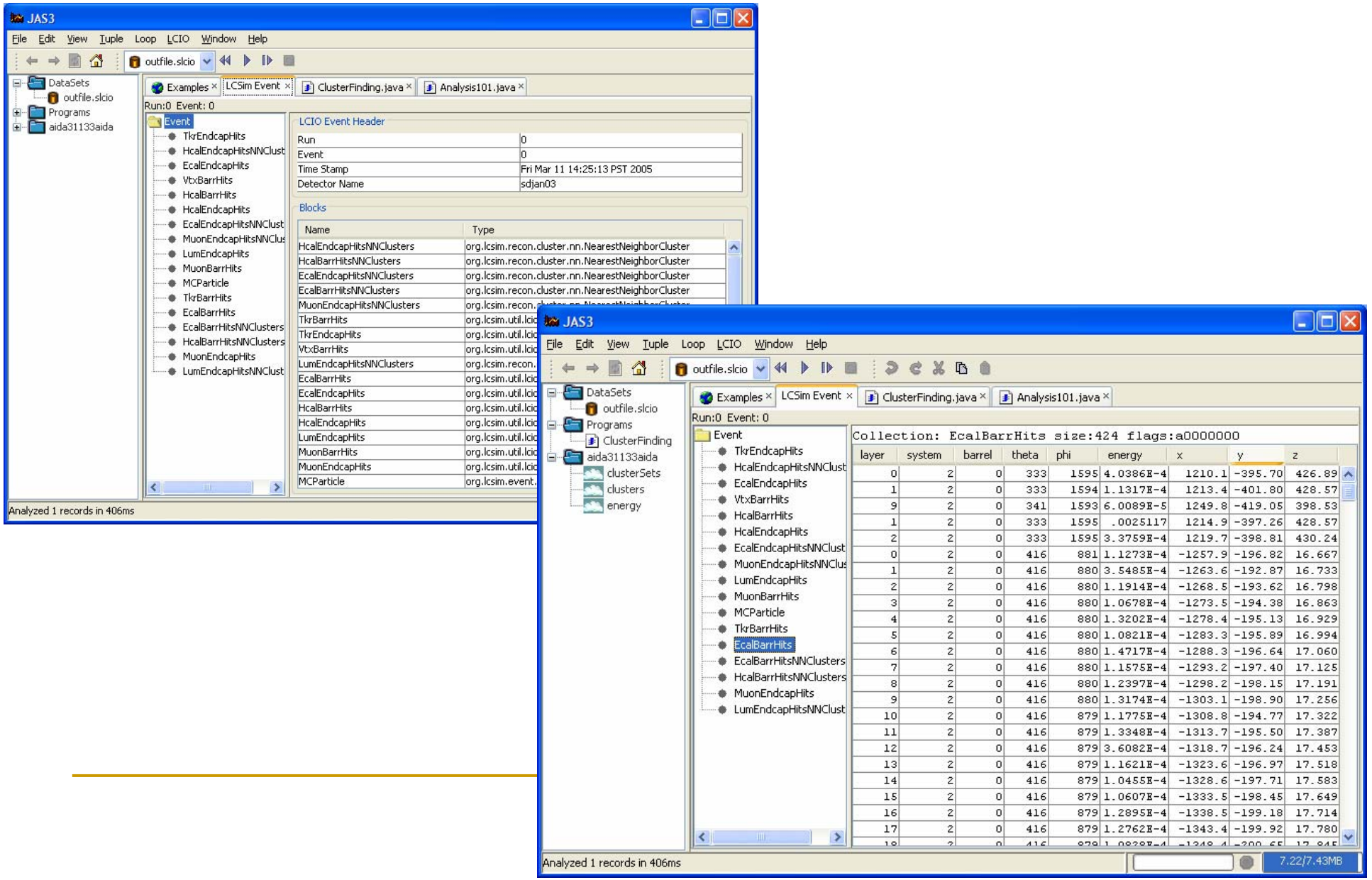
Using org.lcsim with JAS3

- The org.lcsim can be used standalone, withan IDE, or inside JAS3. Same code can be used in all modes, so easy to move back and forth
  - E.g. develop in IDE and run in JAS3
  - E.g. develop in JAS3 and run in batch
- JAS3 org.lcsim plugin adds:
  - Example Analysis Code
  - org.lcim Event browser
  - Easy viewing of analysis plots
  - WIRED event display integration

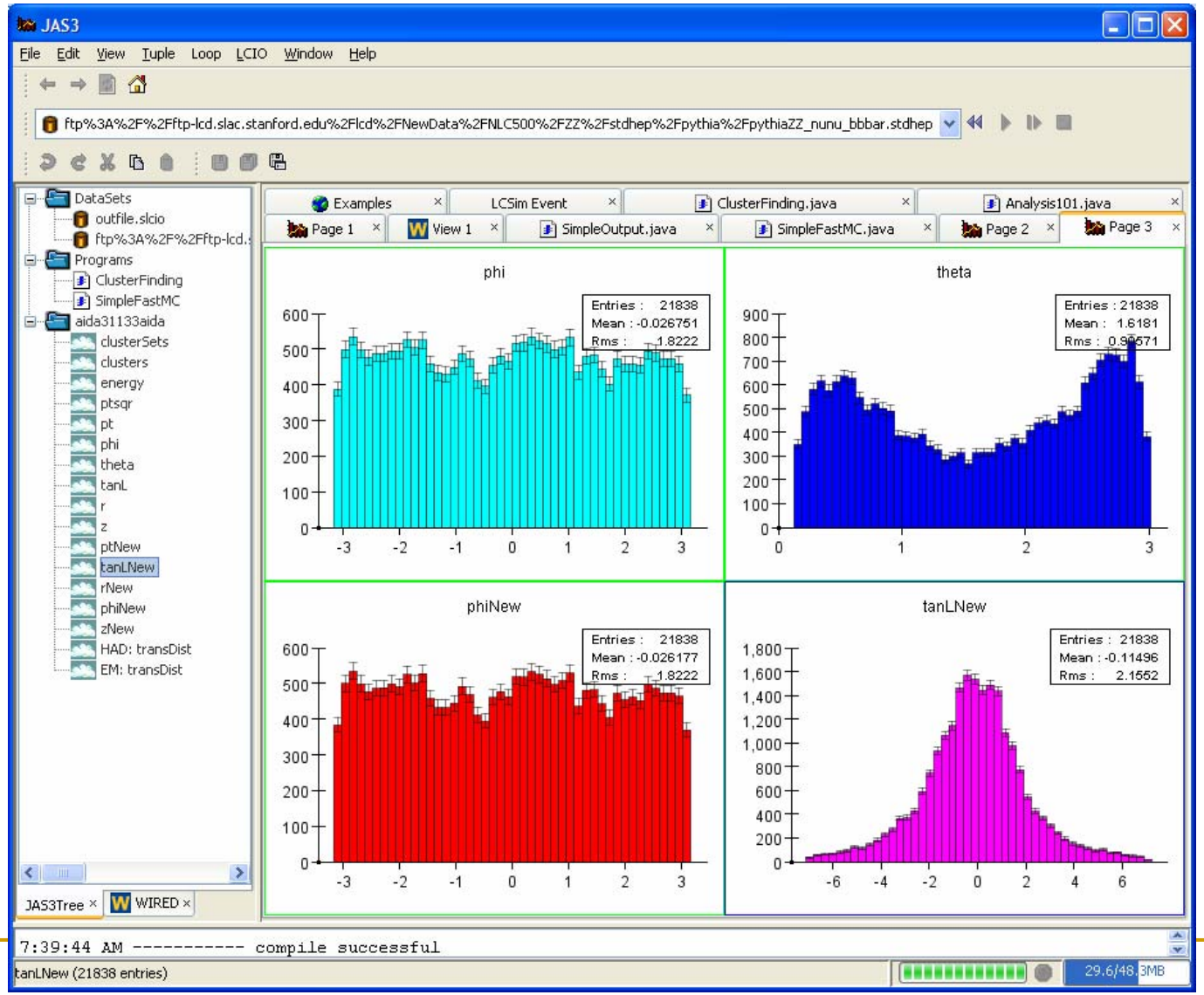
org.lcsim: Examples



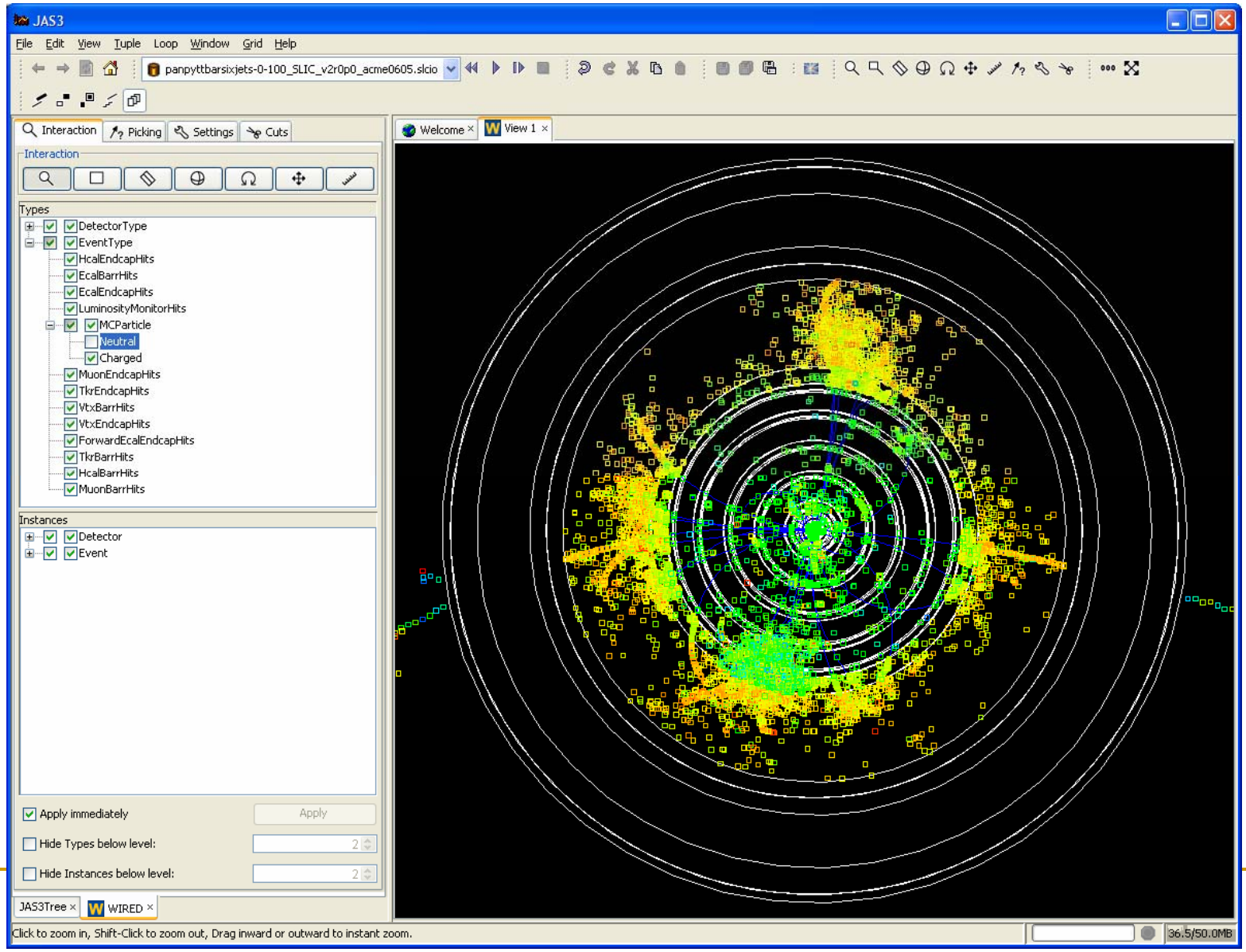
org.lcsim: Examples



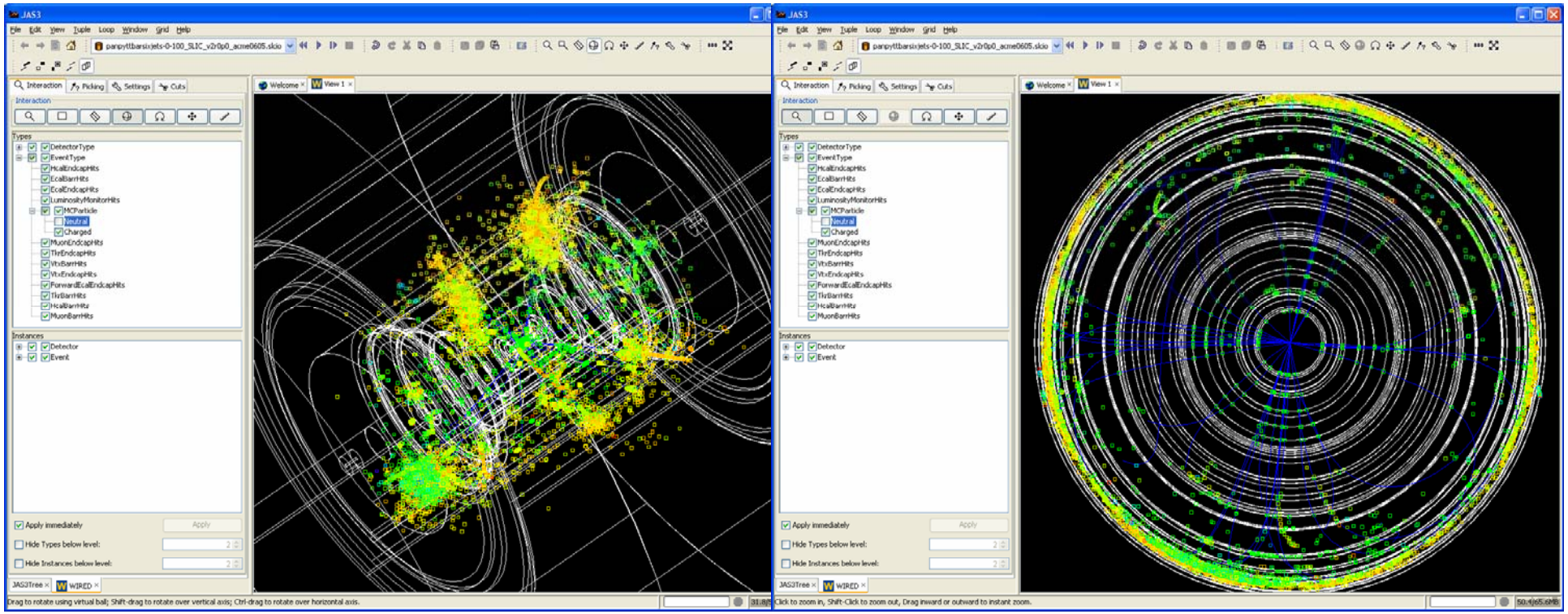
org.lcsim: Plot Viewing



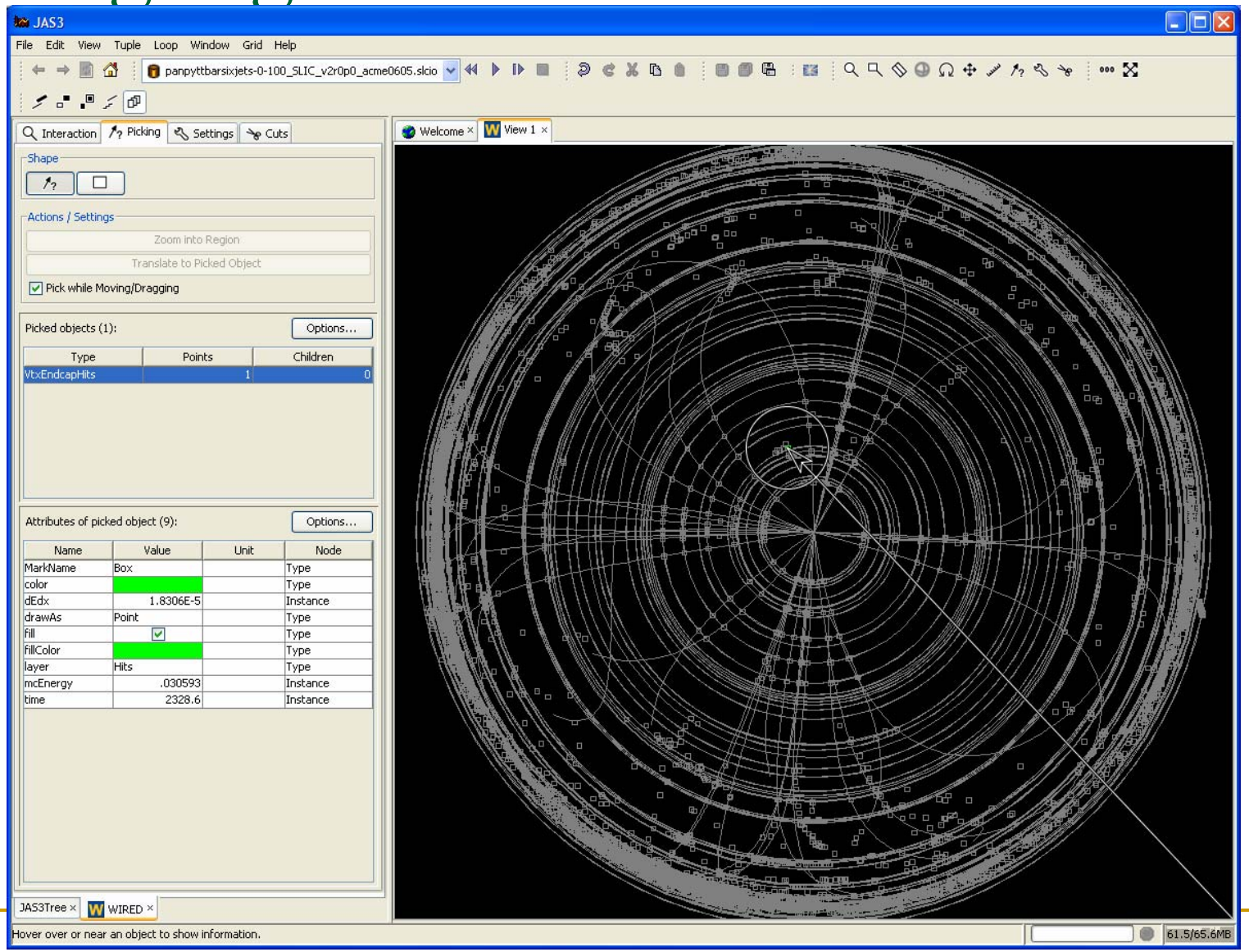
Using org.lcsim with WIRED4



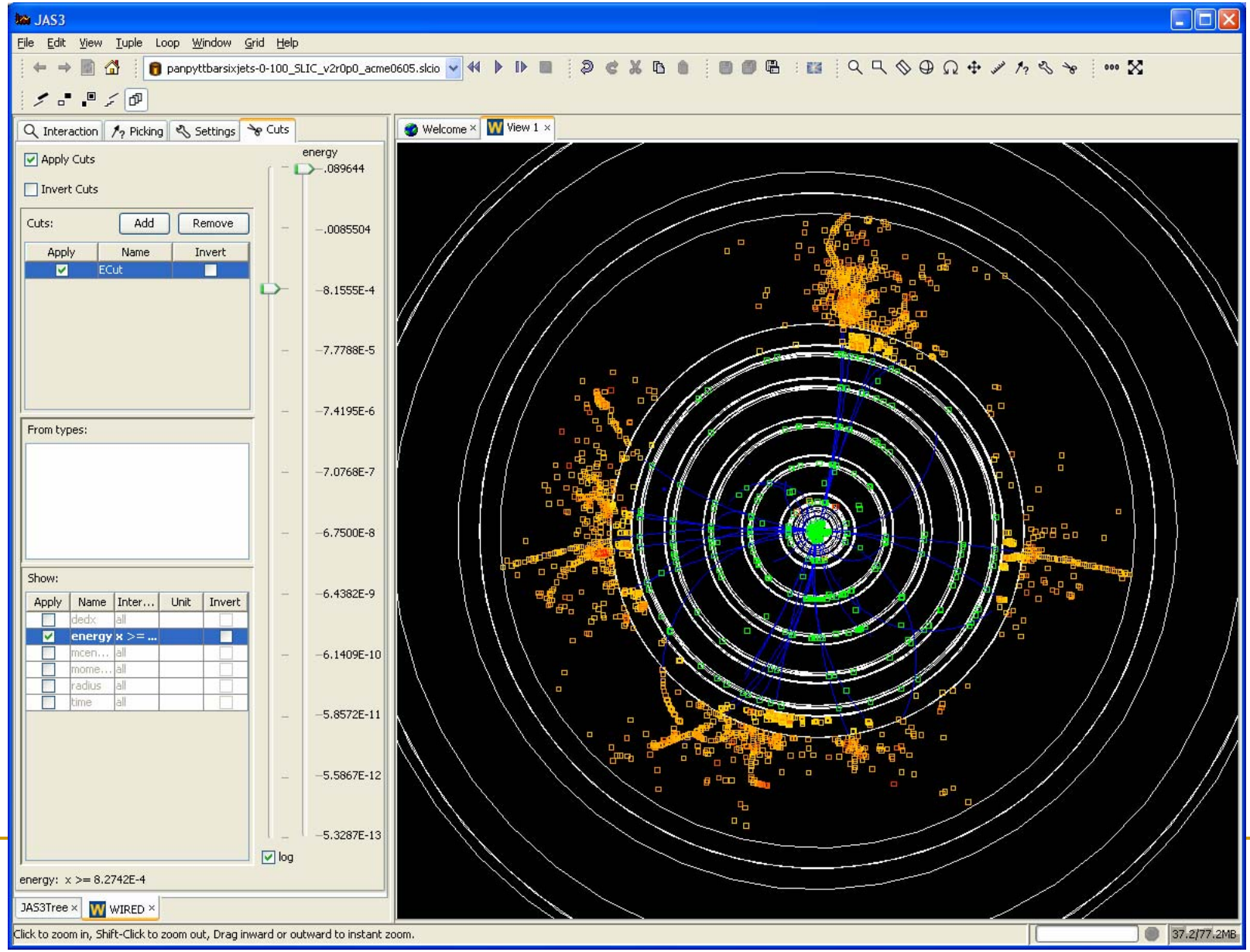
Using org.lcsim with WIRED4



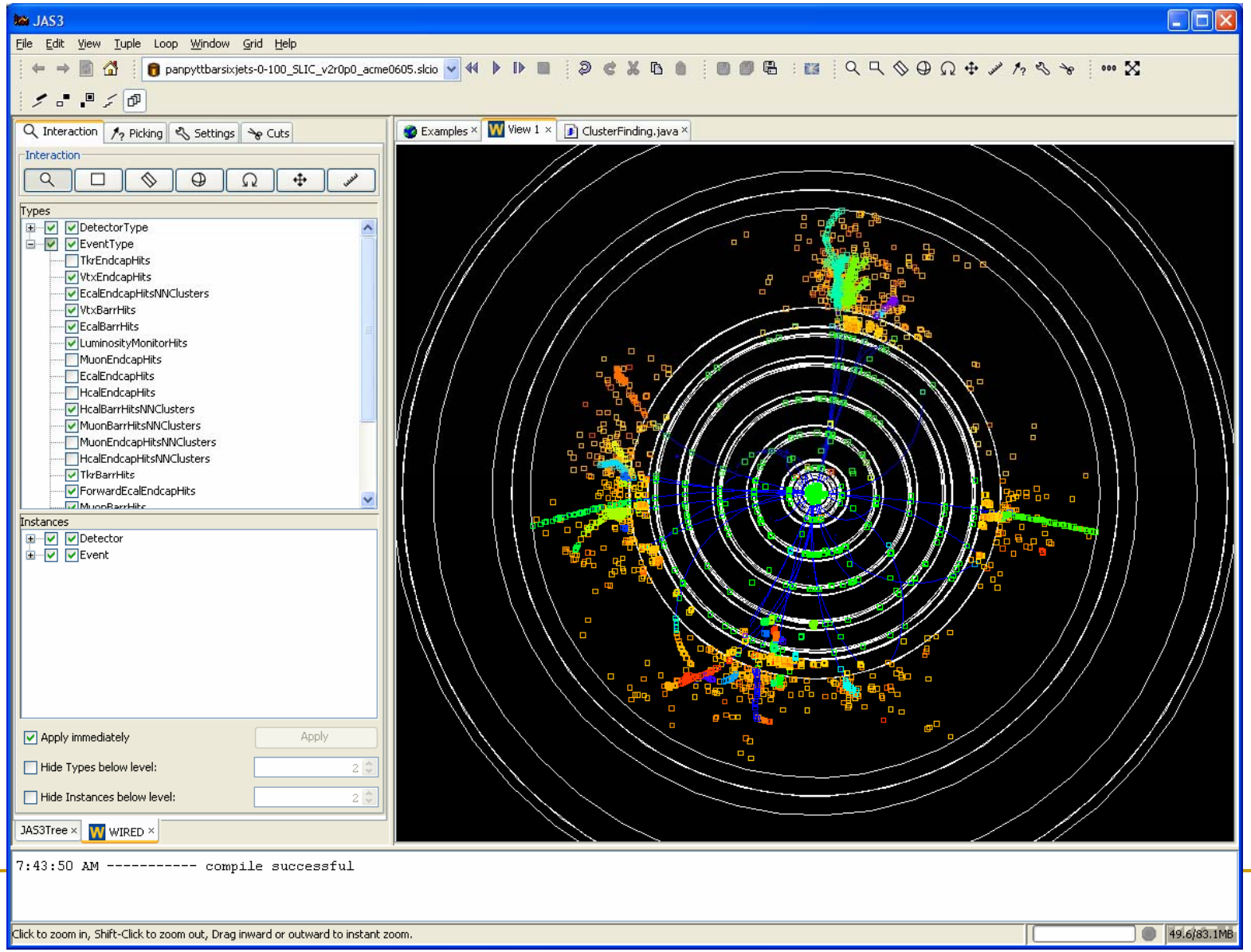
Using org.lcsim with WIRED4



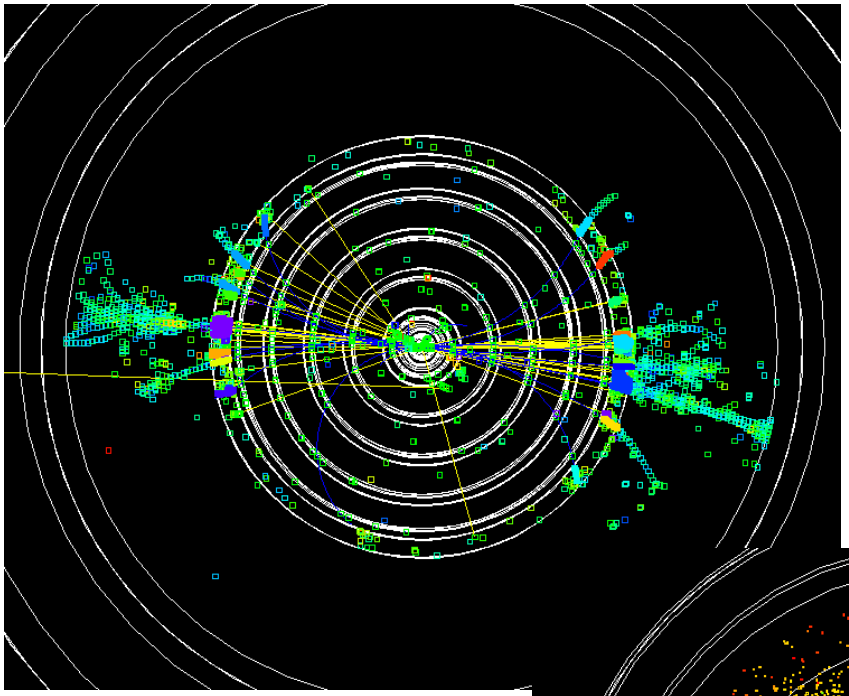
Using org.lcsim with WIRED4



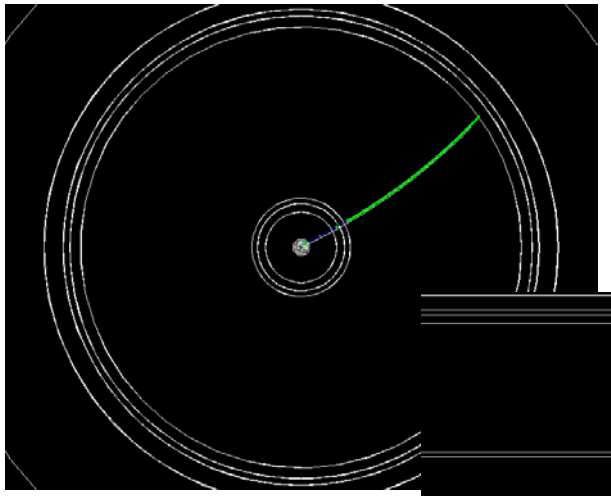
Using org.lcsim with WIRED4



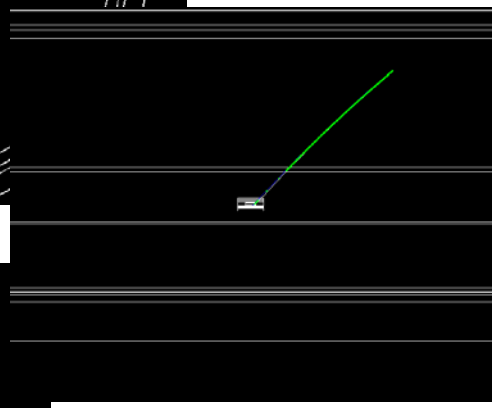
Interoperability

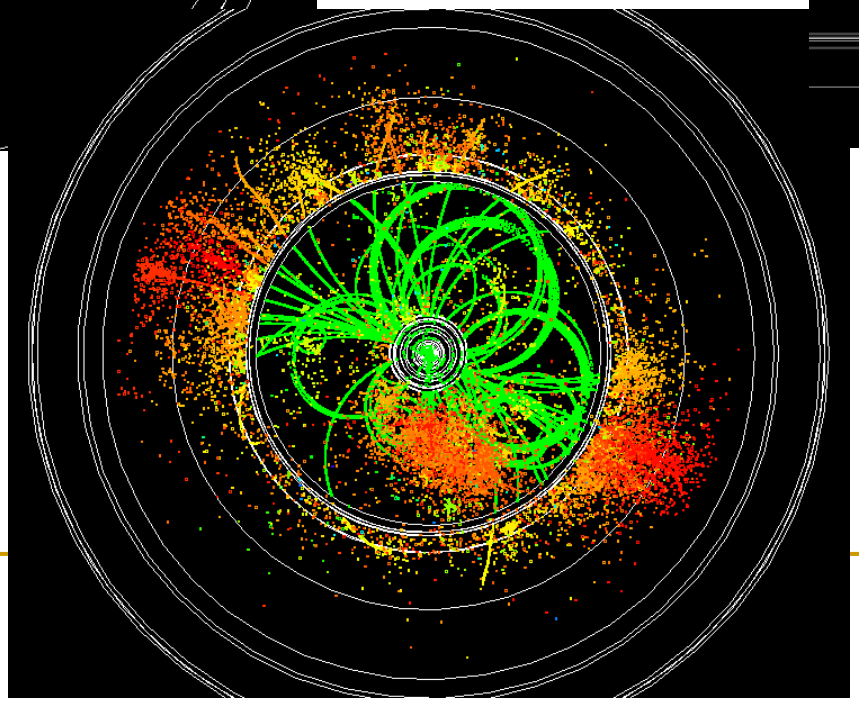


SiD



LDC

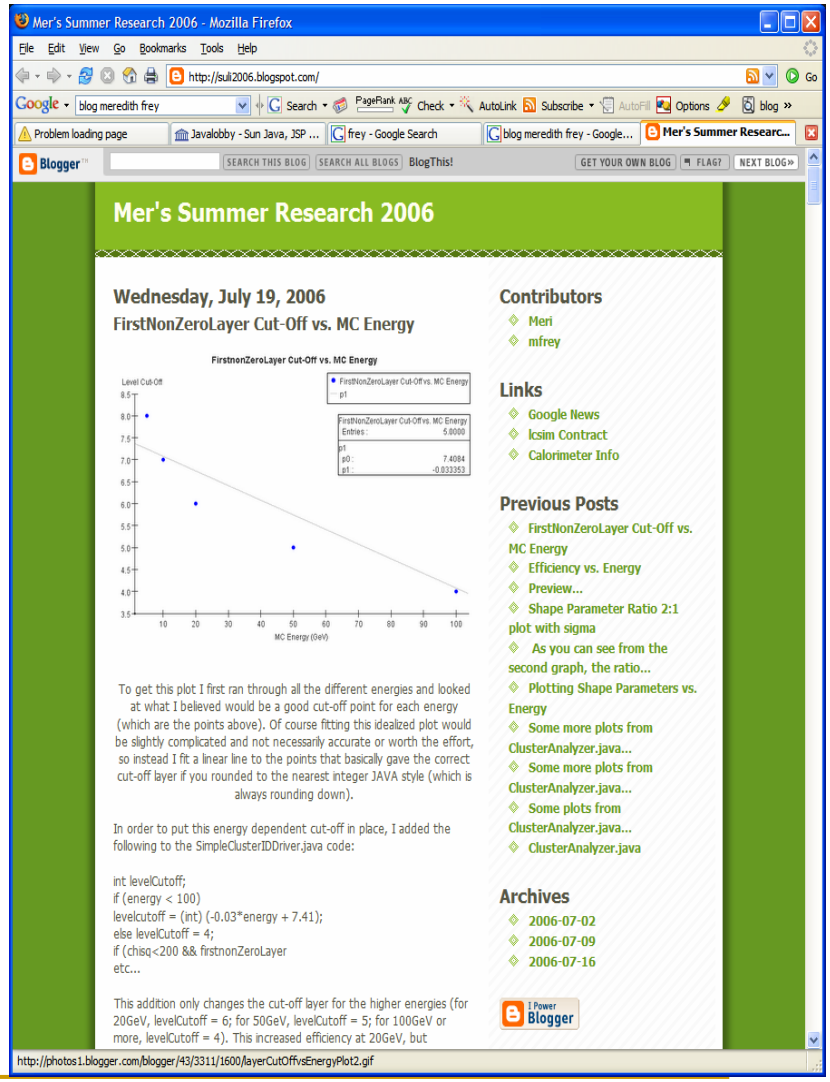




GLD

How hard is it to get started with org.lcsim?

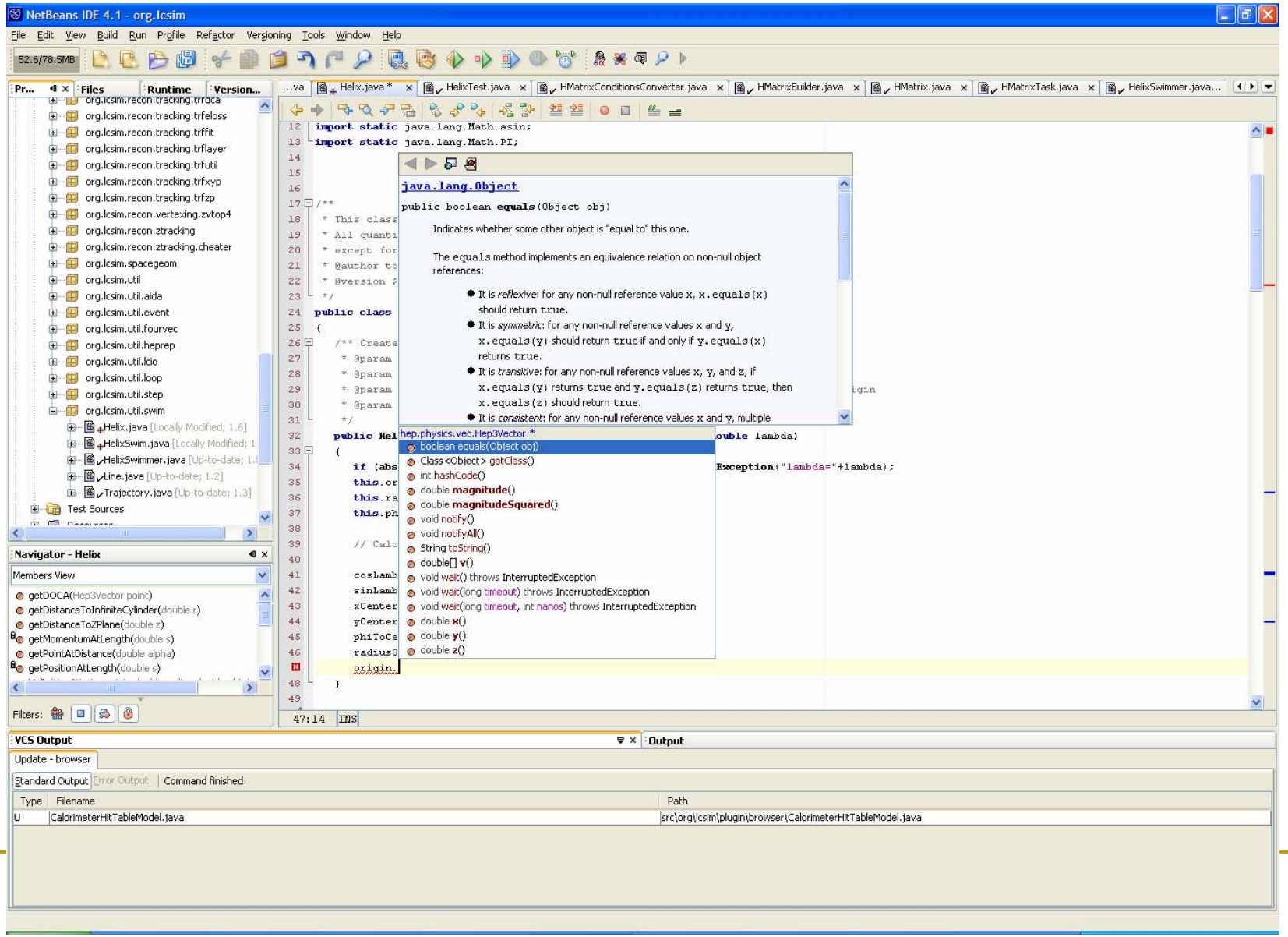
- Works on Linux, MacOSX, Windows
  - Should take about 15 minutes to install JAS3 and org.lcsim plugin.
- Case Study: SLAC Summer student
  - 2 semesters of Java experience
    - (no C++, Fortran *etc*)
  - Using tutorial on lcsim.org Wiki; installed software, downloaded data, and got useful results in one day (and fixed a few errors in the documentation along the way).
  - Regular analysis updates have been appearing on her blog ever since!
- Even if you don't have Java experience you can get started almost as fast
  - (the only thing you will miss is the core dumps)
- Start here:
  - https://confluence.slac.stanford.edu/display/ilc/lcsim+Getting+Started
  - Problems? Attend Tuesday afternoon "Simulation" phone meeting or use discussion forum at http://forum.linearcollider.org/



Becoming an org.lcsim developer

- To get started you just need “Java”, “cvs”, “maven”
  - Maven is a Java based project management tool
  - Single command “maven”
    - downloads dependencies, compiles code, runs tests, deploys code
- All code in CVS
- To check-out and build all code:
  - set CVSROOT=“pserver:anonymous@cvs.freehep.org:/cvs/lcsim”
  - cvs co GeomConverter
  - cd GeomConverter
  - maven
  - cd ..
  - cvs co lcsim
  - cd lcsim
  - maven
- Find more documentation at:
  - http://lcsim.org/
  - Read/Contribute to the Wiki at: https://confluence.slac.stanford.edu/display/ilc/Home
  - Discuss at: http://forum.linearcollider.org/
- We strongly encourage developers to use IDE
  - Netbeans, Eclipse both free, easy to learn, very powerful
  - Use mevenide to teach IDEs about maven systen

Using org.lcsim with Netbeans



Where Next?

- Some clean-up of “Track/Track Parameters” and “Geometry” interface
  - Form org.lcsim clean-up “Task Force”
    - Jeremy and Jan Strube are seconded
    - Other volunteers welcome
- Complete Tracking/Vertexing packages
- Migrate some contrib code to main code base
- We are close to complete tracking/PFA/vertexing/flavor tagging chain.
  - Should create fully simulated/reconstructed data to complement Fast MC studies
- Interoperability
  - LCIO works nicely
  - Geometry interoperability remains elusive but highly desirable
  - Ability to call C++ (MarlinReco) modules from org.lcsim
    - Perhaps more possible with new version of SWIG

Conclusion

- org.lcsim Framework is mostly complete
  - If there are limitations which are impeding your work, let us know!
- User contributed reconstruction software growing rapidly
  - Several more contributions promised soon