org.lcsim Reconstruction and Analysis package

org.lcsim - Contents

- History and Goals
- Geometry System – Compact Detector Description
- Conditions System
- Event Display and Event Browser
- Interoperability with other tools
- Reconstruction Status (Overview)
- Getting Involved

org.lcsim History

- org.lcsim Evolved from
  - *hep.lcd* package in use since last century
    - Full Reconstruction (tracking+calorimetry)
    - FastMC – simple track and cluster smearing
    - Physics Tools (Vertex Finding, Jet Finding)
    - Beam Background Overlays
    - Analysis tools including event display
  - LCIOPlugin
    - LCIO is ILC standard for data storage/exchange
    - Simple tool for viewing/analysing any LCIO file

org.lcsim Goals

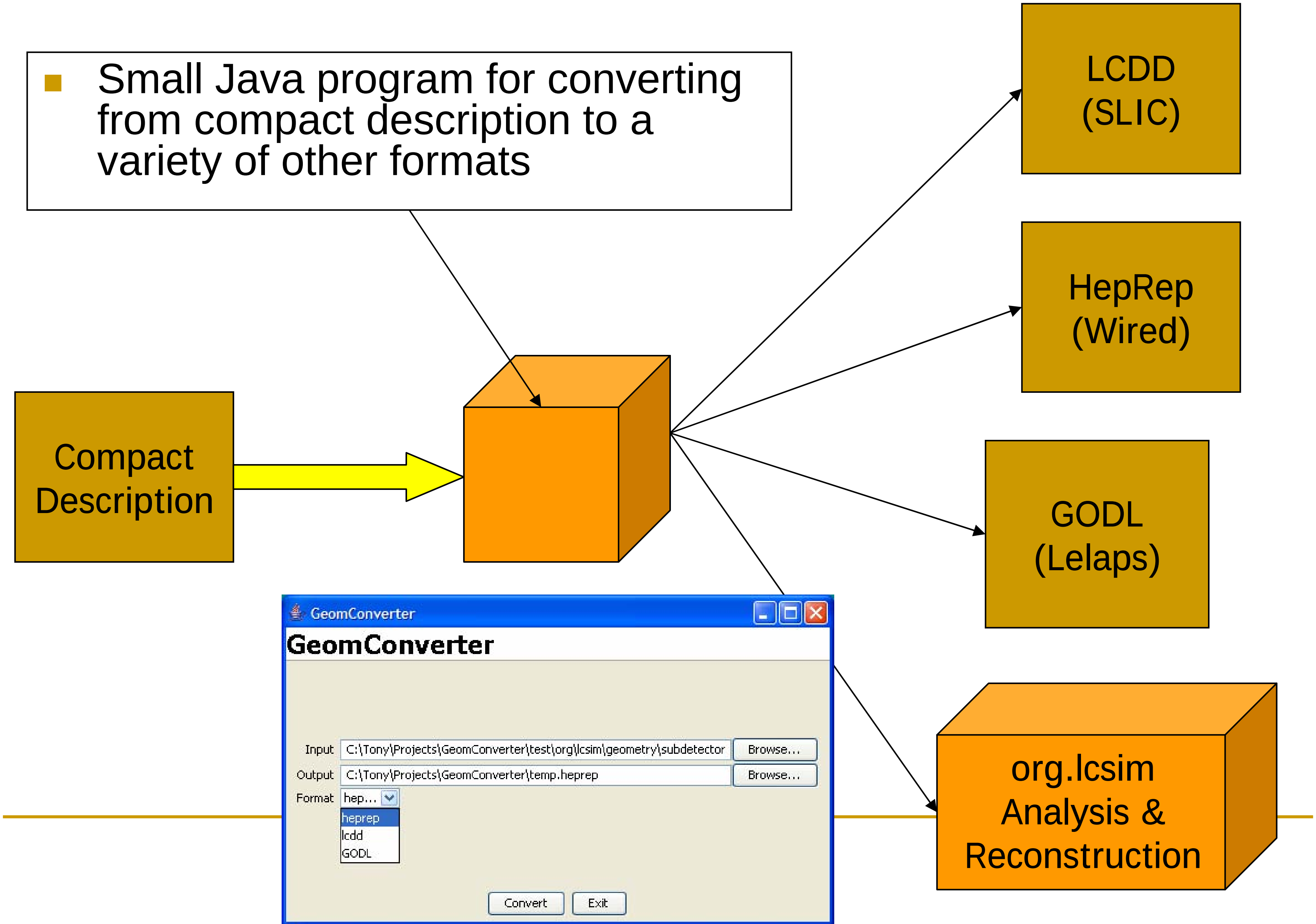
- Retain core functionality from *hep.lcd* package
  - Full suite of reconstruction and analysis tools available to all LCIO users
- Update to use latest LCIO for IO and as basis for simulation, raw data and reconstruction event formats
  - Isolate users from raw LCIO structures
- Update/simplify framework using experience from hep.lcd
  - Provide good tutorial documentation
- Detector Independence
  - Make package independent of detector, geometry assumptions so can work with any detector
  - Read properties of detectors at runtime
- Update to Java 1.5
  - Simple, easy to learn, efficient, OO language
  - Many improvements since hep.lcd framework was created.
- Ability to run standalone (command line or batch) or in JAS3

org.lcsim: Compact Geometry Description

- LCDD is great, handles any geometry, but
  - Files are large, since entry for every G4 volume
  - Simple change (e.g. # layers) may require many changes to LCDD file
  - Not right level of detail for reconstruction
- Compact format is less generic, but
  - Files are much shorter and easier to edit
  - Can handle any likely geometry/segmentation
    - May require additional “drivers” to be implemented in Java
  - Maintains XML advantages
  - LCDD can be generated from compact format
- Goal:
  - Rapid prototyping of detector geometries
  - Ability to provide description of new (or existing) detectors for reconstruction (org.lcsim)

org.lcsim: Geometry Converter

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



GeomConverter

Input C:\Tony\Projects\GeomConverter\test\org\lcsim\geometry\subdetector

Output C:\Tony\Projects\GeomConverter\temp.heprep

Format hep...

org.lcsim Conditions Database

- Goal: Using “detector name” in LCIO file
  - Provide access to a extensible set of conditions including:
    - Detector Geometry
    - Algorithm Specific Constants
      - E.g. FastMC smearing parameters
  - Don’t make assumptions about format of data
  - Don’t rely on internet access, or local database installation
  - Make it easy for users to define new detectors

org.lcsim Conditions Database: Implementation

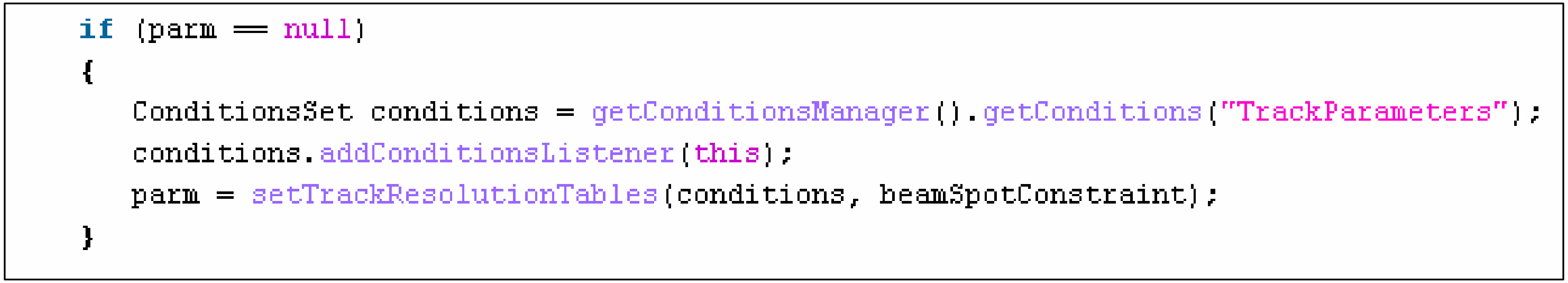
- Detector Constants stored in .zip file
  - Typically contains:
    - Compact geometry file
    - Set of (ascii) constants for standard algorithms
  - Can additionally contain:
    - Arbitrary files (xml, ascii, binary) needed by other algorithms
    - Other geometry formats (HepRep, LCDD)
    - Full fieldmap
- To define a new detector just create a new .zip file.

org.lcsim Conditions Database: Implementation

- org.lcsim locates file based on detector name by:
  - Looking in alias.properties file in
    - User’s home directory
    - Built-in to org.lcsim software
    - On www.lcsim.org website*
  - If translated name is a URL (file:, ftp:, http)
    - Read .zip file at specified location
  - Search for .zip file in
    - ~/.lcsim/detectors
    - Built-in to org.lcsim software
    - In http://www.lcsim.org/detectors*
      - *Downloaded files are cached locally for future use

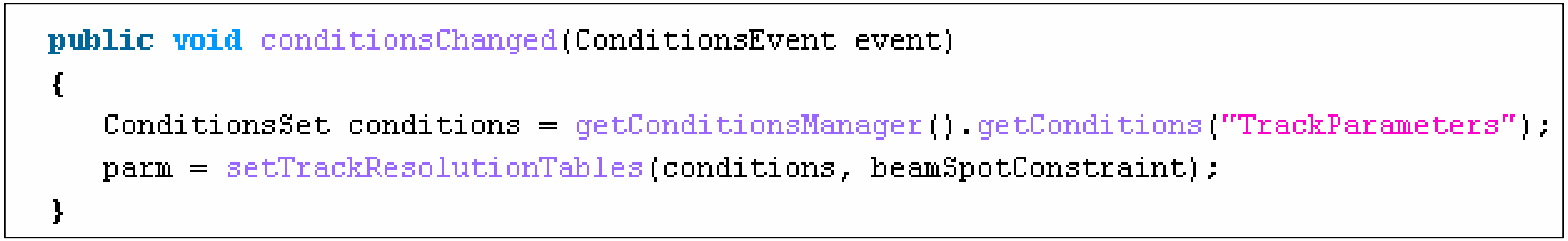
org.lcsim Conditions Database: Accessing constants

- Constants are accessed by name (actually path in zip file)



```
    if (parm == null)
    {
       ConditionsSet conditions = getConditionsManager().getConditions("TrackParameters");
       conditions.addConditionsListener(this);
       parm = setTrackResolutionTables(conditions, beamSpotConstraint);
    }
```

- Later on if constants change, listener will be called



```
  public void conditionsChanged(ConditionsEvent event)
  {
     ConditionsSet conditions = getConditionsManager().getConditions("TrackParameters");
     parm = setTrackResolutionTables(conditions, beamSpotConstraint);
  }
```

Geometry Access

- Geometry Access is build on top of conditions database
  - Framework reads and decodes compact geometry description.
    - User can iterate over detectors or access detector by name or id
      - Provides access to detector properties (# layers etc).
      - “Shape” of detector
  - Framework builds “IDDecoder” objects for each hit collection.
    - ID decoder unpacks ID’s in LCIO files
      - Provides access hit position, detector layer etc.
      - Can provide additional utilities, for instance list of neighboring cells in calorimeter

Available Detector Descriptions

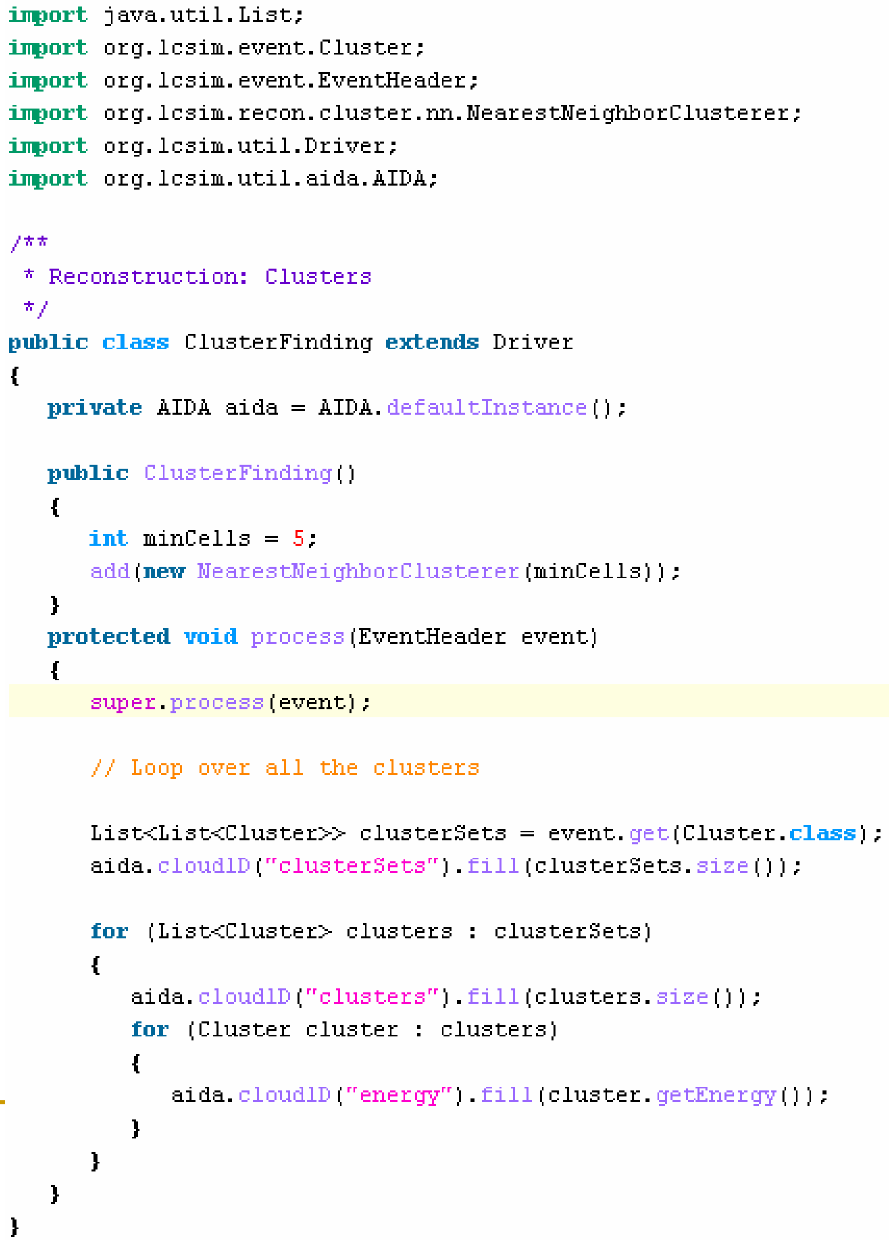
- Although detector descriptions can live anywhere we maintain a CVS repository of detector descriptions
  - Exported to org.lcsim web site for automatic download
- Many SiD variants
  - sdjan03, sdfeb05, sidaug05, sidaug05_4tesla, sidaug05_20mr, sidaug05_np, sidaug05_scinthcal, sidmay05, sidmay05_2mr, sidmay05_20mr, sidmay05_np, sidmay05_scinthcal
- Other Detectors
  - gldaug05, ldcaug05, cdcaug05
- You are welcome to contribute more

org.lcsim Drivers, Event Access

- Reconstruction and Analysis Code is written by extending Driver class.
  - Most code extends only a single method
    - process(EventHeader event)
  - which passes in EventHeader through which all event data, conditions, and geometry is accessed.
- EventHeader maintains features which users liked from old hep.lcd framework, but adds compatibility with LCIO events
  - All LCIO data is accessible.
  - Arbitrary user data, either collections or single objects can be added to event.
  - Event can be written out
    - Currently only objects understood by LCIO are written out
      - MCParticles, Hits, Clusters, Tracks, Reconstructed Particles (can include Jets, Vertices)
      - Can use LCIO “GenericObjects” to write out more

Reconstruction: Clustering

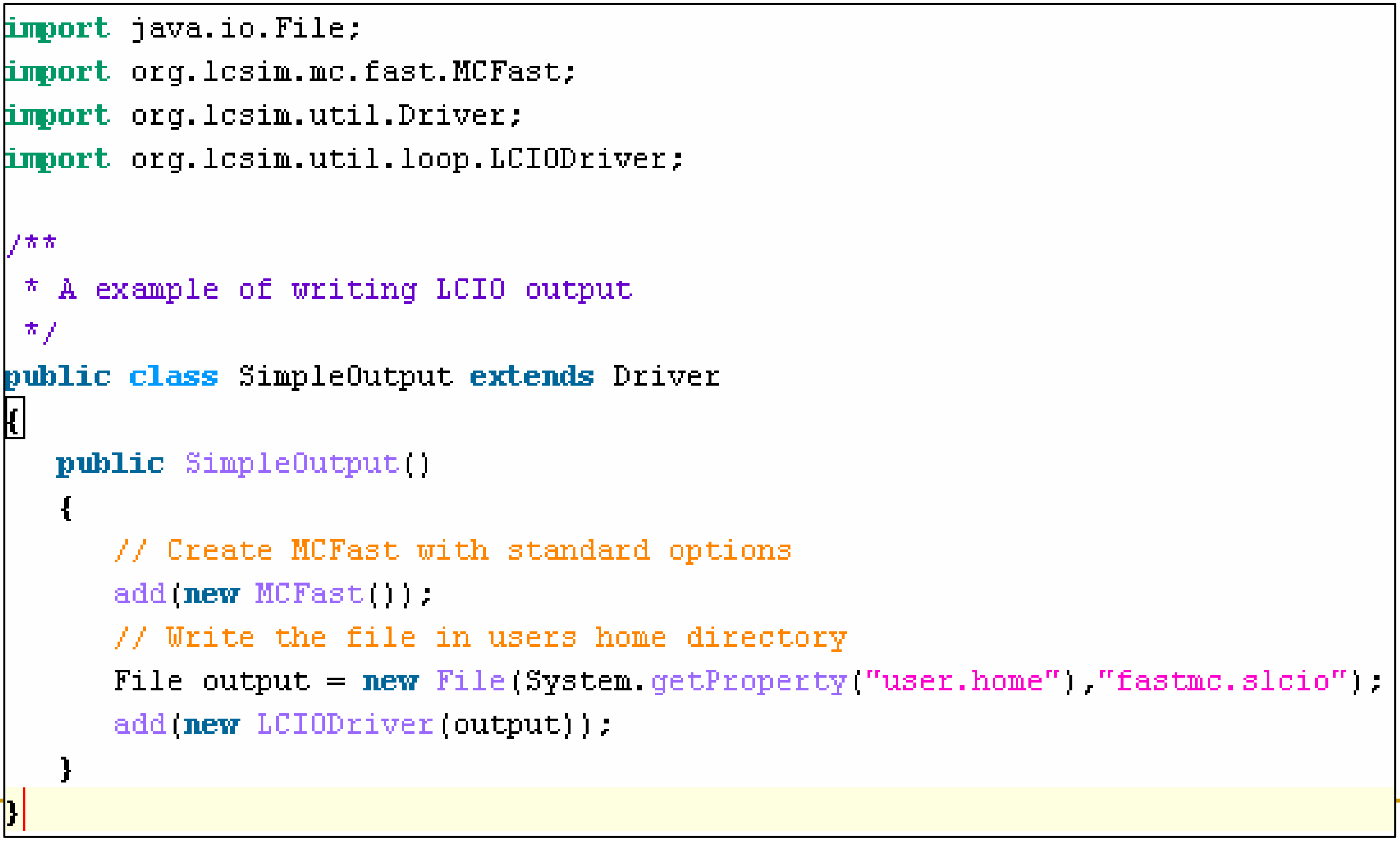
- Several Clustering Algorithms now included
  - Cluster Cheater
    - uses MC information
  - Nearest Neighbor Clusterer
  - Cone Clusterer
- Algorithms can be run in parallel for comparison



```java
import java.util.List;
import org.lcsim.event.Cluster;
import org.lcsim.event.EventHeader;
import org.lcsim.recon.cluster.nn.NearestNeighborClusterer;
import org.lcsim.util.Driver;
import org.lcsim.util.aida.AIDA;

/**
 * Reconstruction: Clusters
 */
public class ClusterFinding extends Driver
{
   private AIDA aida = AIDA.defaultInstance();

   public ClusterFinding()
   {
      int minCells = 5;
      add(new NearestNeighborClusterer(minCells));
   }
   protected void process(EventHeader event)
   {
      super.process(event);

      // Loop over all the clusters

      List<List<Cluster>> clusterSets = event.get(Cluster.class);
      aida.cloud1D("clusterSets").fill(clusterSets.size());

      for (List<Cluster> clusters : clusterSets)
      {
         aida.cloud1D("clusters").fill(clusters.size());
         for (Cluster cluster : clusters)
         {
            aida.cloud1D("energy").fill(cluster.getEnergy());
         }
      }
   }
}
```

org.lcsim: IO

- Can read LCIO files, stdhep files (for FastMC)
- Can write LCIO files

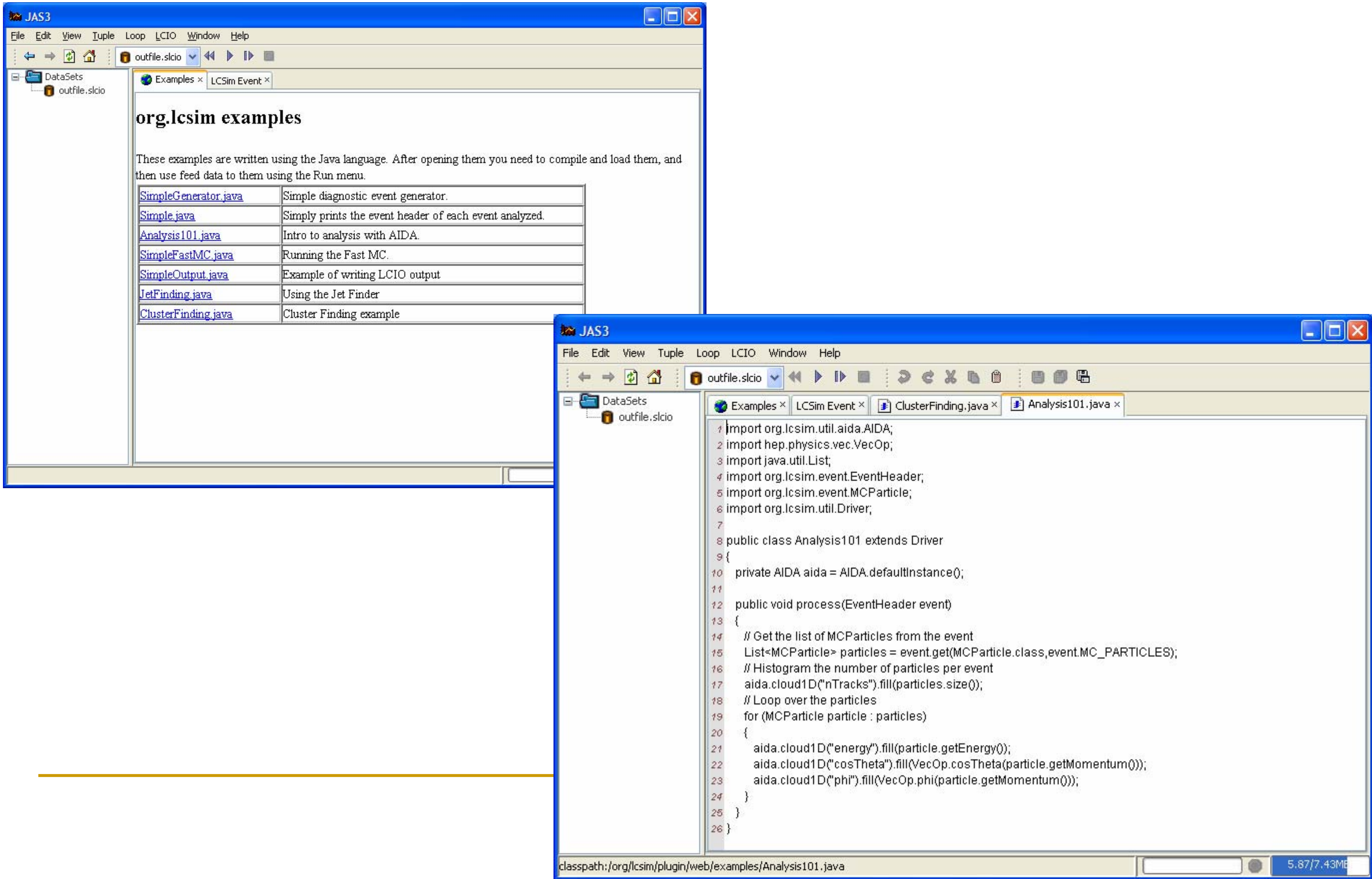


```java
import java.io.File;
import org.lcsim.mc.fast.MCFast;
import org.lcsim.util.Driver;
import org.lcsim.util.loop.LCIODriver;

/**
 * A example of writing LCIO output
 */
public class SimpleOutput extends Driver
{
   public SimpleOutput()
   {
      // Create MCFast with standard options
      add(new MCFast());
      // Write the file in users home directory
      File output = new File(System.getProperty("user.home"),"fastmc.slcio");
      add(new LCIODriver(output));
   }
}
```

org.lcsim: JAS Plugin

- The org.lcsim can be used standalone, or inside JAS3. Same code can be used in both places, 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 plugin adds:
  - Example Analysis Code
  - Event browser (similar to LCIOPlugin)
  - Easy viewing of AIDA plots
  - WIRED event display integration

org.lcsim: Examples



org.lcsim examples

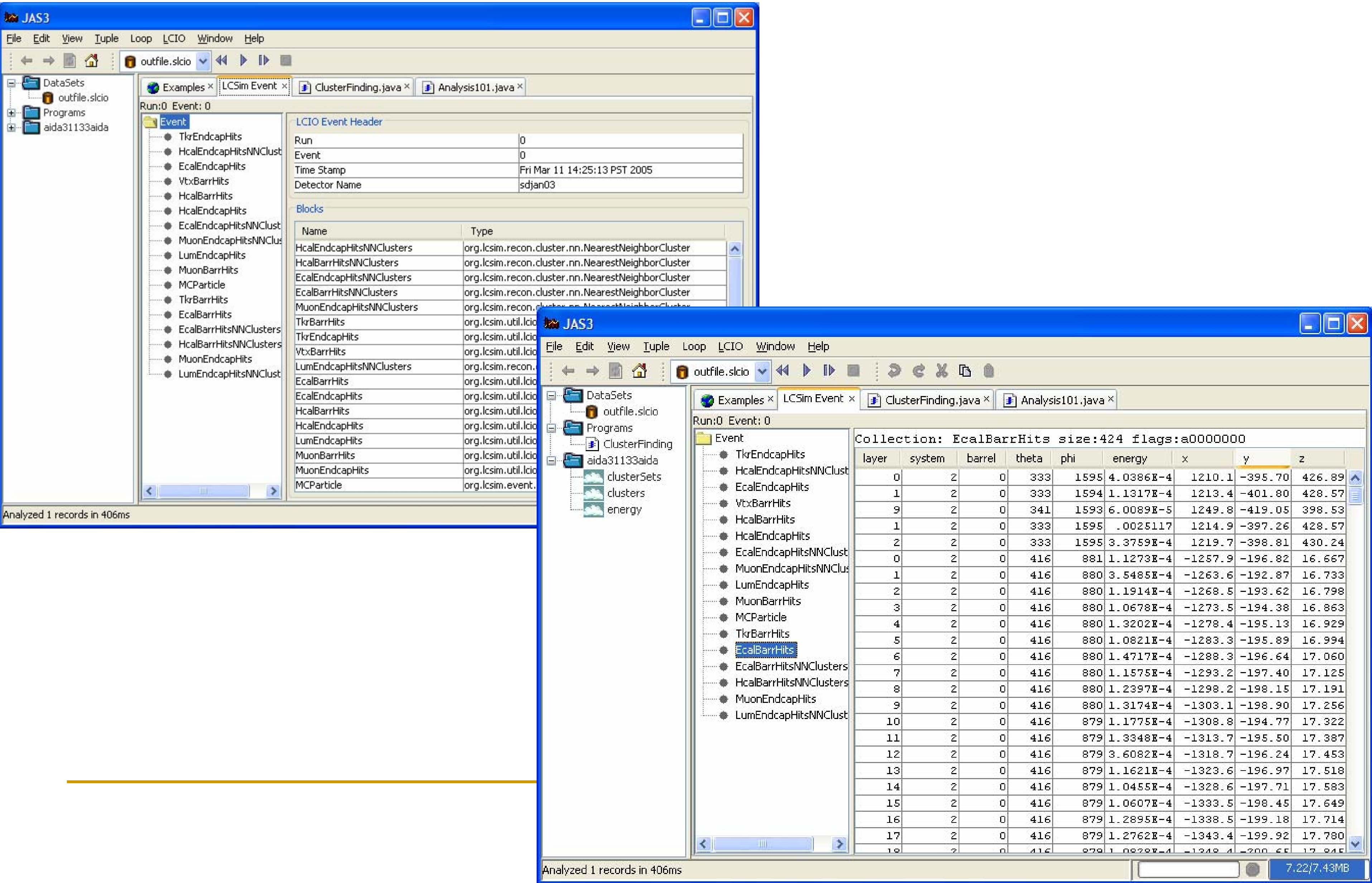
These examples are written using the Java language. After opening them you need to compile and load them, and then use feed data to them using the Run menu.

SimpleGenerator.java	Simple diagnostic event generator.
Simple.java	Simply prints the event header of each event analyzed.
Analysis101.java	Intro to analysis with AIDA.
SimpleFastMC.java	Running the Fast MC.
SimpleOutput.java	Example of writing LCIO output
JetFinding.java	Using the Jet Finder
ClusterFinding.java	Cluster Finding example

```
import org.lcsim.util.aida.AIDA;
import hep.physics.vec.VecOp;
import java.util.List;
import org.lcsim.event.EventHeader;
import org.lcsim.event.MCParticle;
import org.lcsim.util.Driver;

public class Analysis101 extends Driver
{
  private AIDA aida = AIDA.defaultInstance();

  public void process(EventHeader event)
  {
    // Get the list of MCParticles from the event
    List<MCParticle> particles = event.get(MCParticle.class,event.MC_PARTICLES);
    // Histogram the number of particles per event
    aida.cloud1D("nTracks").fill(particles.size());
    // Loop over the particles
    for (MCParticle particle : particles)
    {
      aida.cloud1D("energy").fill(particle.getEnergy());
      aida.cloud1D("cosTheta").fill(VecOp.cosTheta(particle.getMomentum()));
      aida.cloud1D("phi").fill(VecOp.phi(particle.getMomentum()));
    }
  }
}
```

classpath:/org/lcsim/plugin/web/examples/Analysis101.java

org.lcsim: Examples



Run:0 Event: 0

LCIO Event Header

Run	0
Event	0
Time Stamp	Fri Mar 11 14:25:13 PST 2005
Detector Name	sdjan03

Blocks

Name	Type
HcalEndcapHitsNNClusters	org.lcsim.recon.cluster.nn.NearestNeighborCluster
HcalBarrHitsNNClusters	org.lcsim.recon.cluster.nn.NearestNeighborCluster
EcalEndcapHitsNNClusters	org.lcsim.recon.cluster.nn.NearestNeighborCluster
EcalBarrHitsNNClusters	org.lcsim.recon.cluster.nn.NearestNeighborCluster
MuonEndcapHitsNNClusters	org.lcsim.recon.cluster.nn.NearestNeighborCluster
TkrBarrHits	org.lcsim.util.lcio
TkrEndcapHits	org.lcsim.util.lcio
VtxBarrHits	org.lcsim.util.lcio
LumEndcapHitsNNClusters	org.lcsim.recon.
EcalBarrHits	org.lcsim.util.lcio
EcalEndcapHits	org.lcsim.util.lcio
HcalBarrHits	org.lcsim.util.lcio
HcalEndcapHits	org.lcsim.util.lcio
LumEndcapHits	org.lcsim.util.lcio
MuonBarrHits	org.lcsim.util.lcio
MuonEndcapHits	org.lcsim.util.lcio
MCParticle	org.lcsim.event.

Analyzed 1 records in 406ms

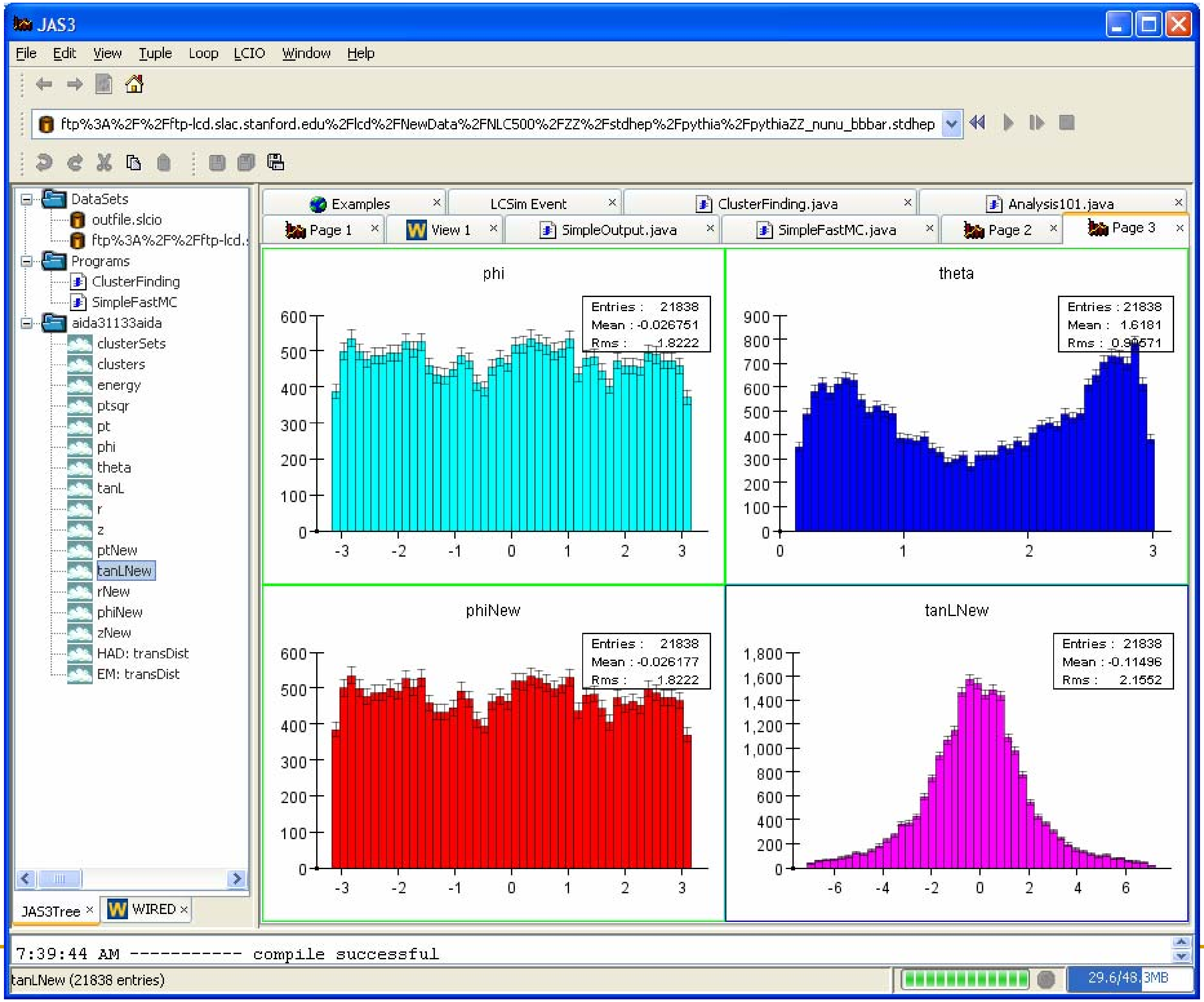
Run:0 Event: 0

Collection: EcalBarrHits size:424 flags:a0000000

layer	system	barrel	theta	phi	energy	x	y	z
0	2	0	333	1595	4.0386E-4	1210.1	-395.70	426.89
1	2	0	333	1594	1.1317E-4	1213.4	-401.80	428.57
9	2	0	341	1593	6.0089E-5	1249.8	-419.05	398.53
1	2	0	333	1595	.0025117	1214.9	-397.26	428.57
2	2	0	333	1595	3.3759E-4	1219.7	-398.81	430.24
0	2	0	416	881	1.1273E-4	-1257.9	-196.82	16.667
1	2	0	416	880	3.5485E-4	-1263.6	-192.87	16.733
2	2	0	416	880	1.1914E-4	-1268.5	-193.62	16.798
3	2	0	416	880	1.0678E-4	-1273.5	-194.38	16.863
4	2	0	416	880	1.3202E-4	-1278.4	-195.13	16.929
5	2	0	416	880	1.0821E-4	-1283.3	-195.89	16.994
6	2	0	416	880	1.4717E-4	-1288.3	-196.64	17.060
7	2	0	416	880	1.1575E-4	-1293.2	-197.40	17.125
8	2	0	416	880	1.2397E-4	-1298.2	-198.15	17.191
9	2	0	416	880	1.3174E-4	-1303.1	-198.90	17.256
10	2	0	416	879	1.1775E-4	-1308.8	-194.77	17.322
11	2	0	416	879	1.3348E-4	-1313.7	-195.50	17.387
12	2	0	416	879	3.6082E-4	-1318.7	-196.24	17.453
13	2	0	416	879	1.1621E-4	-1323.6	-196.97	17.518
14	2	0	416	879	1.0455E-4	-1328.6	-197.71	17.583
15	2	0	416	879	1.0607E-4	-1333.5	-198.45	17.649
16	2	0	416	879	1.2895E-4	-1338.5	-199.18	17.714
17	2	0	416	879	1.2762E-4	-1343.4	-199.92	17.780
18	2	0	416	879	1.0978E-4	-1348.4	-200.65	17.845

Analyzed 1 records in 406ms

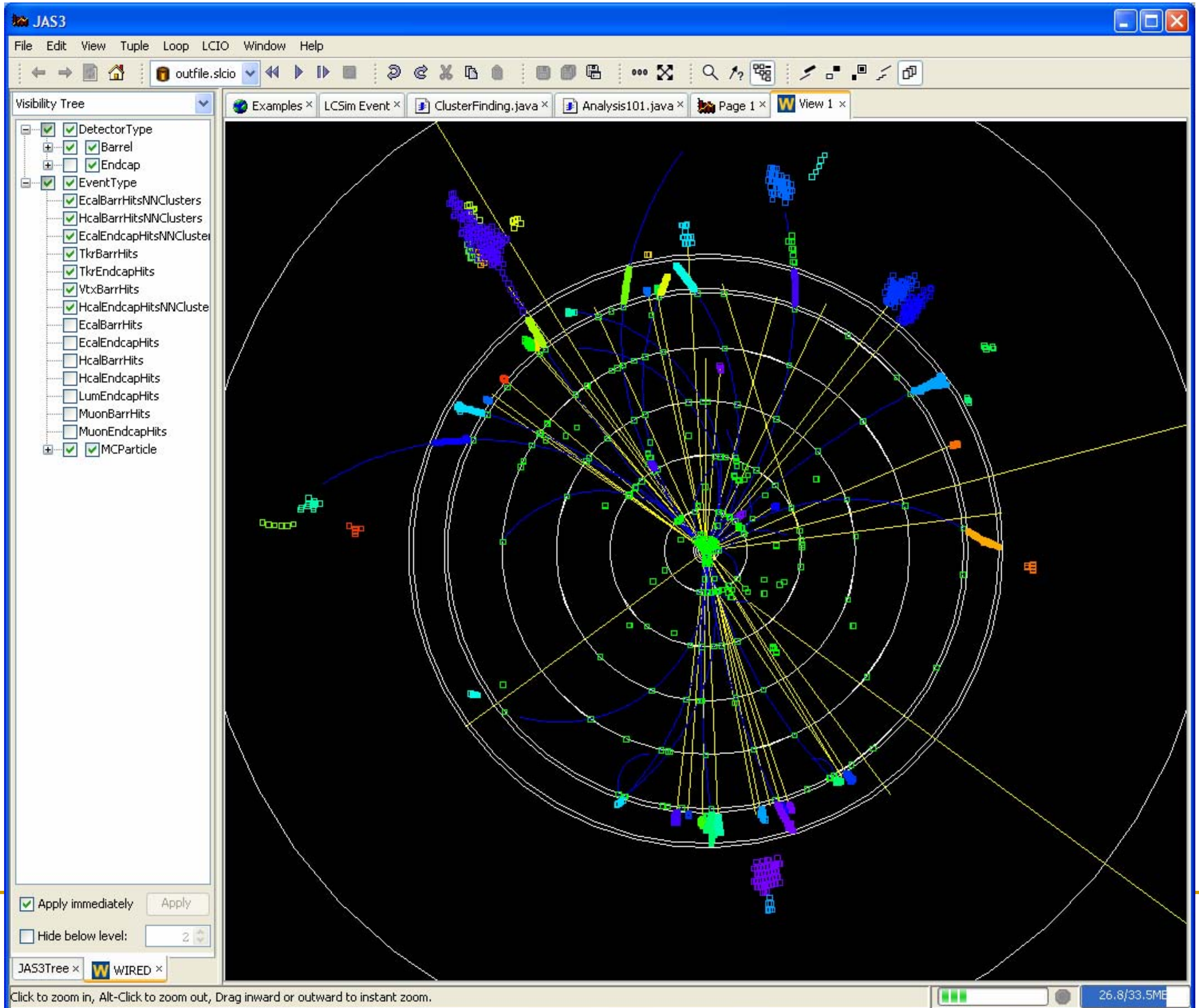
org.lcsim: Plot Viewing



```
7:39:44 AM ----------- compile successful
```

tanLNew (21838 entries)

Org.lcsim: WIRED event display



Visibility Tree

- ☑ ☑ DetectorType
  - ☑ ☑ Barrel
  - □ ☑ Endcap
- ☑ ☑ EventType
  - ☑ EcalBarrHitsNNClusters
  - ☑ HcalBarrHitsNNClusters
  - ☑ EcalEndcapHitsNNCluster
  - ☑ TkrBarrHits
  - ☑ TkrEndcapHits
  - ☑ VtxBarrHits
  - ☑ HcalEndcapHitsNNCluste
  - □ EcalBarrHits
  - □ EcalEndcapHits
  - □ HcalBarrHits
  - □ HcalEndcapHits
  - □ LumEndcapHits
  - □ MuonBarrHits
  - □ MuonEndcapHits
  - ☑ ☑ MCParticle

☑ Apply immediately   Apply

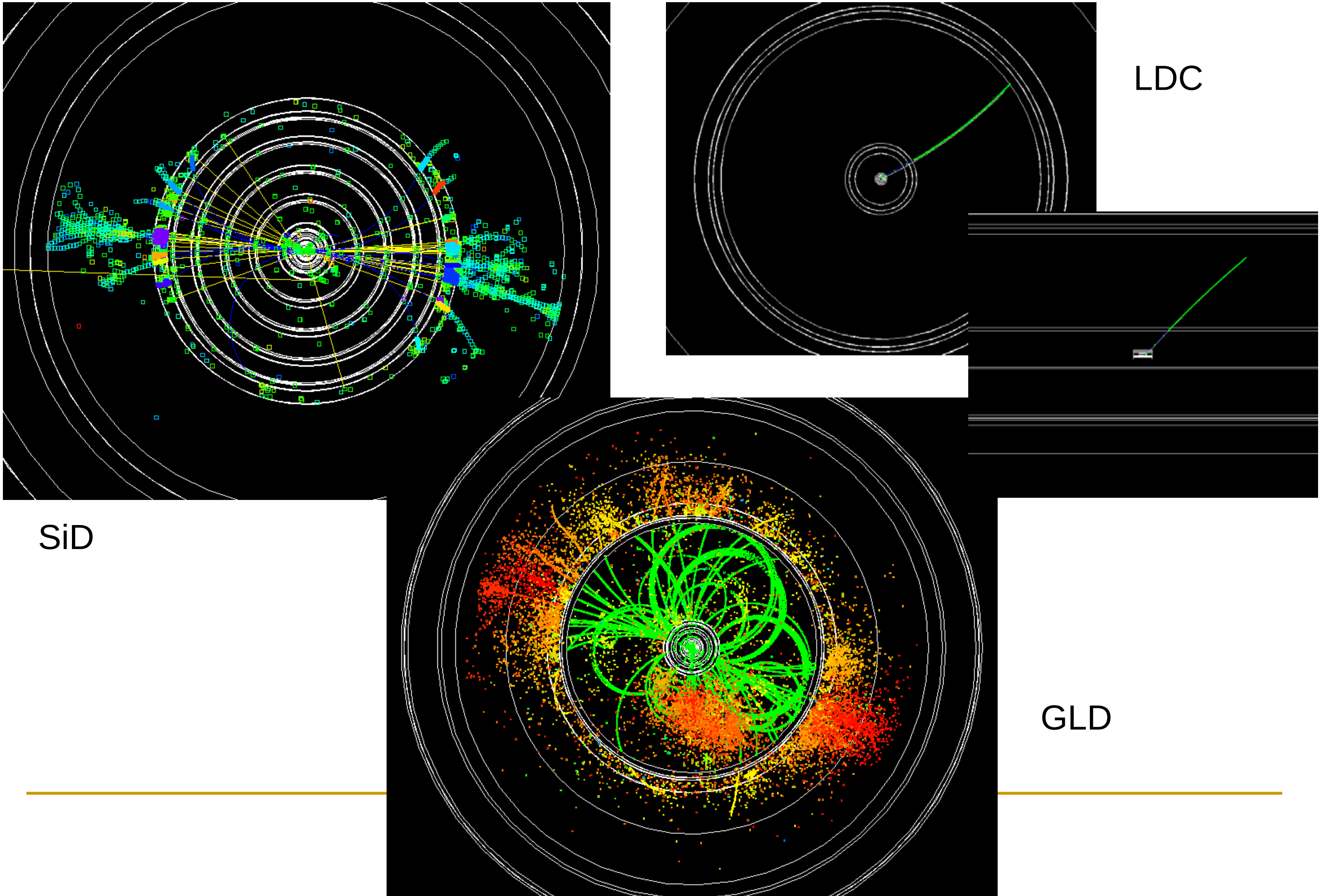
□ Hide below level: 2

Click to zoom in, Alt-Click to zoom out, Drag inward or outward to instant zoom.

org.lcsim: Interoperability

- Working during this workshop to improve interoperability
  - Will no longer require Compact Geometry Description for
    - Event browser (some functionality will be reduced)
    - WIRED event display
      - Can provide HepRep and/or field strength
      - Ability to optionally suppress hits from “loopers” in TPC
  - Individual reconstruction algorithms will vary on their requirements

Interoperability



org.lcsim Status

- Physics Utilities - done
  - stdhep reader
  - 3, 4-vector utilities
  - diagnostic generator
  - Jet finder, event shape utilities
- Conditions framework – done
  - Ability to read detector constants from “zip” file
  - To define new detector just create new zip file and place on web
    - File is read and cached locally
  - Ability to read compact geometry file
- Driver framework – done
- Fast MC – done
- IO Framework – done
- Event Access – done
- Event Display interface – done
- Reconstruction
  - In progress

Reconstruction Status

- Goal of org.lcsim is not to produce a single reconstruction program, but to provide a framework into which many algorithms can be included.
  - **FastMC**
    - Simple parameterized track and cluster smearing
      - Smearing constants read from conditions system
    - Now produces ReconstructedParticles
  - **Digitization**
    - Digisim – Calorimeter digitization
    - *Tracker, Vertex digitization*
  - **Clustering**
    - Cheater, Cone, Nearest Neighbor, *Minimal Spanning Tree*
  - **Tracking**
    - SLDWeightMatrix, TRF, Garfield
  - **Vertex Fitting**
    - ZVTop4
  - **Analysis Examples**
    - Cluster Diagnostics, SLICDiagnostics, PFA, ClusterID, ParticleID
  - For more details see individual talks in physics sessions at this workshop

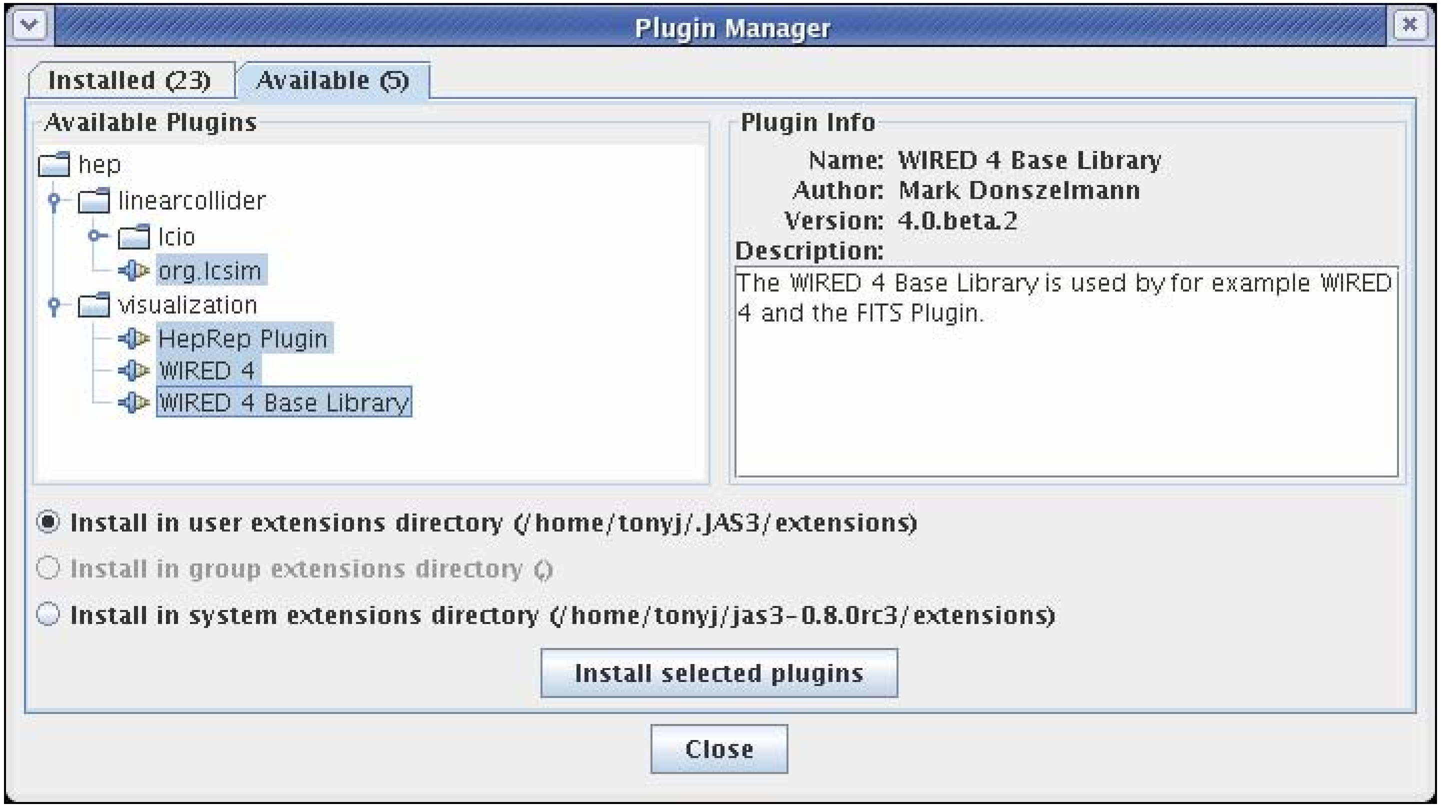
Contributors

- Contributors Include (in random order):
  - T.Johnson, N. Graf, R. Cassell, J. McCormick, S. Magill, J. Strube, M. Ronan, M. Charles, C. Milstene, F. Gaede, M. Donszelmann, G. Bower, M. Turri, V. Serbo, R. Partridge, N. Sinev, W. Walkowiak, W. Mader, T. Nelson, S. Kuhlman, G. Lima, B. Schumm, D. Onoprienko, W. Langfeld, N. Sinev, …
  - <Your-name-here>

How to Contribute

- Getting started is very easy
  - Install JAS3 from Snowmass CD or from web
    - Windows, Linux, MacOSX (>=10.4)
  - Use the JAS3 Plugin Manager to install the org.lcsim plugin
  - Use the event display, event browser
  - Run the built-in examples
  - Develop your own analysis
- Instructions: http://confluence.slac.stanford.edu//x/VxE

JAS3 plugin installation



Name: WIRED 4 Base Library
Author: Mark Donszelmann
Version: 4.0.beta.2
Description:
The WIRED 4 Base Library is used by for example WIRED 4 and the FITS Plugin.

● Install in user extensions directory (/home/tonyj/.JAS3/extensions)

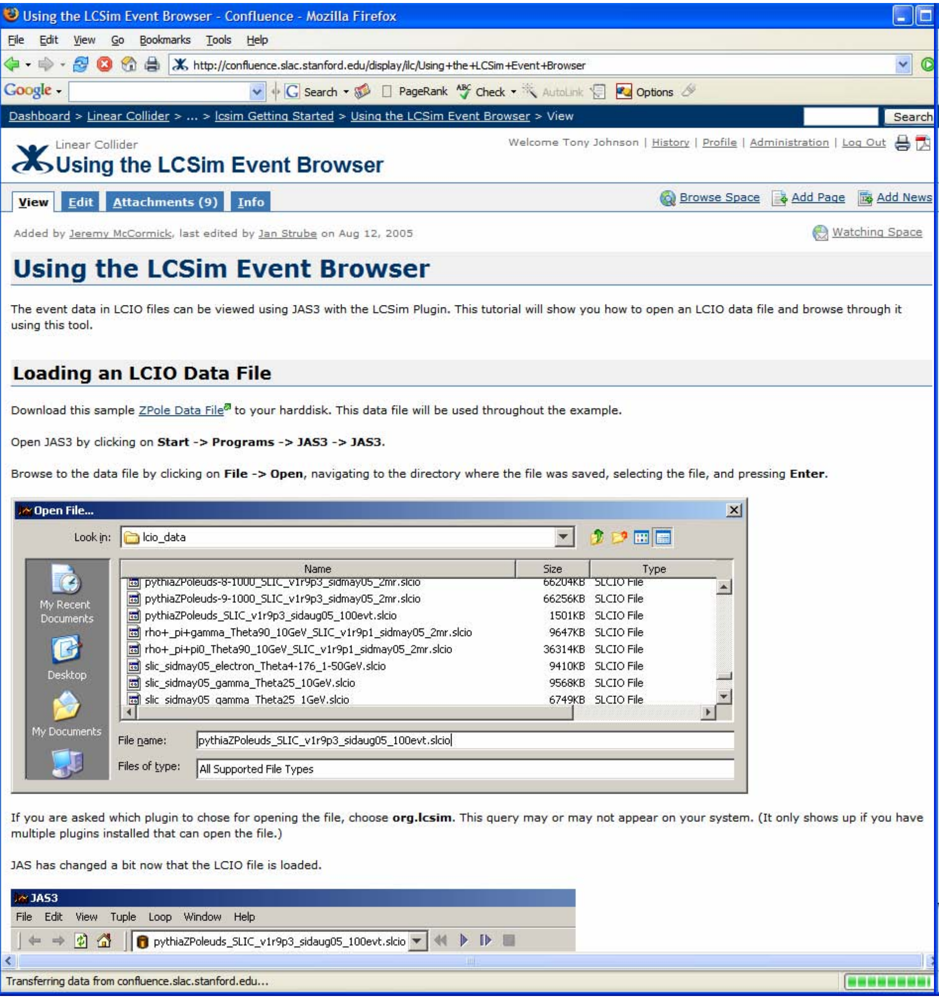
○ Install in group extensions directory ()

○ Install in system extensions directory (/home/tonyj/jas3-0.8.0rc3/extensions)

How to contribute, continued

- Documentation
  - Main web site is http://lcsim.org/
  - Most documentation is stored in *“Confluence”* Wiki
    - http://confluence.slac.stanford.edu/display/ilc/Home
    - Anyone can (and is encouraged to) add comments
    - Anyone can get an account and improve documentation
      - Confluence is very easy to use,
        - just click “Edit” link using any web browser
        - instructions are given on side bar of editor page
  - JIRA bug tracking system is used to keep track of problems and plans.
    - http://jira.slac.stanford.edu/

Confluence



Using the LCSim Event Browser - Confluence - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://confluence.slac.stanford.edu/display/ilc/Using+the+LCSim+Event+Browser
Google Search PageRank Check AutoLink Options
Dashboard > Linear Collider > ... > lcsim Getting Started > Using the LCSim Event Browser > View
Search
Linear Collider
Using the LCSim Event Browser
Welcome Tony Johnson | History | Profile | Administration | Log Out
View Edit Attachments (9) Info
Browse Space Add Page Add News
Watching Space

Added by Jeremy McCormick, last edited by Jan Strube on Aug 12, 2005

Using the LCSim Event Browser

The event data in LCIO files can be viewed using JAS3 with the LCSim Plugin. This tutorial will show you how to open an LCIO data file and browse through it using this tool.

Loading an LCIO Data File

Download this sample ZPole Data File to your harddisk. This data file will be used throughout the example.

Open JAS3 by clicking on **Start -> Programs -> JAS3 -> JAS3.**

Browse to the data file by clicking on **File -> Open**, navigating to the directory where the file was saved, selecting the file, and pressing **Enter**.

Name	Size	Type
pythiaZPoleuds-8-1000_SLIC_v1r9p3_sidmay05_2mr.slcio	66204KB	SLCIO File
pythiaZPoleuds-9-1000_SLIC_v1r9p3_sidmay05_2mr.slcio	66256KB	SLCIO File
pythiaZPoleuds_SLIC_v1r9p3_sidaug05_100evt.slcio	1501KB	SLCIO File
rho+_pi+gamma_Theta90_10GeV_SLIC_v1r9p1_sidmay05_2mr.slcio	9647KB	SLCIO File
rho+_pi+pi0_Theta90_10GeV_SLIC_v1r9p1_sidmay05_2mr.slcio	36314KB	SLCIO File
slic_sidmay05_electron_Theta4-176_1-50GeV.slcio	9410KB	SLCIO File
slic_sidmay05_gamma_Theta25_10GeV.slcio	9568KB	SLCIO File
slic_sidmay05_gamma_Theta25_1GeV.slcio	6749KB	SLCIO File

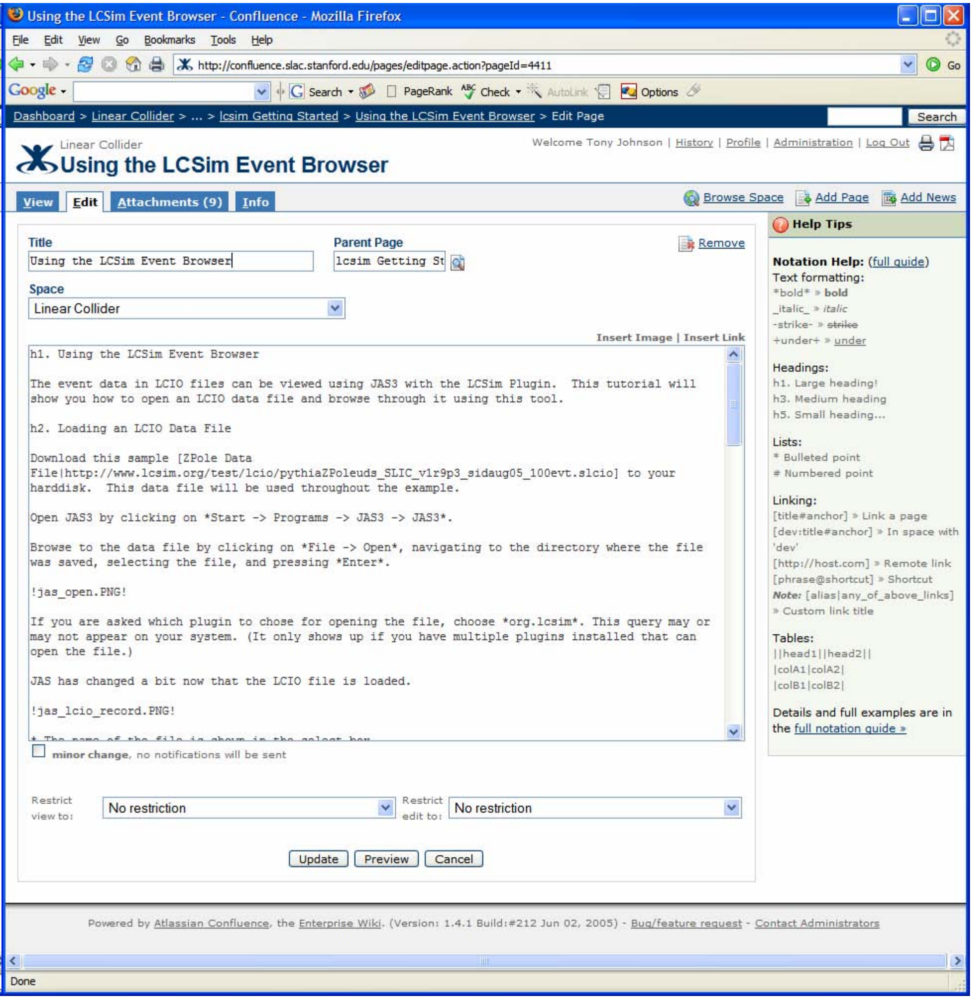
File name: pythiaZPoleuds_SLIC_v1r9p3_sidaug05_100evt.slcio

Files of type: All Supported File Types

If you are asked which plugin to chose for opening the file, choose **org.lcsim**. This query may or may not appear on your system. (It only shows up if you have multiple plugins installed that can open the file.)

JAS has changed a bit now that the LCIO file is loaded.

JAS3
File Edit View Tuple Loop Window Help
pythiaZPoleuds_SLIC_v1r9p3_sidaug05_100evt.slcio
Transferring data from confluence.slac.stanford.edu...



Using the LCSim Event Browser - Confluence - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://confluence.slac.stanford.edu/pages/editpage.action?pageId=4411
Go
Google Search PageRank Check AutoLink Options
Dashboard > Linear Collider > ... > lcsim Getting Started > Using the LCSim Event Browser > Edit Page
Search
Linear Collider
Using the LCSim Event Browser
Welcome Tony Johnson | History | Profile | Administration | Log Out
View Edit Attachments (9) Info
Browse Space Add Page Add News
Remove
Insert Image | Insert Link

Title: Using the LCSim Event Browser

Parent Page: lcsim Getting St

Space: Linear Collider

```
h1. Using the LCSim Event Browser

The event data in LCIO files can be viewed using JAS3 with the LCSim Plugin.  This tutorial will
show you how to open an LCIO data file and browse through it using this tool.

h2. Loading an LCIO Data File

Download this sample [ZPole Data
File|http://www.lcsim.org/test/lcio/pythiaZPoleuds_SLIC_v1r9p3_sidaug05_100evt.slcio] to your
harddisk.  This data file will be used throughout the example.

Open JAS3 by clicking on *Start -> Programs -> JAS3 -> JAS3*.

Browse to the data file by clicking on *File -> Open*, navigating to the directory where the file
was saved, selecting the file, and pressing *Enter*.

!jas_open.PNG!

If you are asked which plugin to chose for opening the file, choose *org.lcsim*. This query may or
may not appear on your system. (It only shows up if you have multiple plugins installed that can
open the file.)

JAS has changed a bit now that the LCIO file is loaded.

!jas_lcio_record.PNG!

* The name of the file is shown in the select box
```

☐ **minor change**, no notifications will be sent

Restrict view to: No restriction

Restrict edit to: No restriction

Help Tips

Notation Help: (full guide)

Text formatting:
bold » **bold**
italic » *italic*
-strike- » ~~strike~~
+under+ » under

Headings:
h1. Large heading!
h3. Medium heading
h5. Small heading...

Lists:
* Bulleted point
# Numbered point

Linking:
[title#anchor] » Link a page
[dev:title#anchor] » In space with 'dev'
[http://host.com] » Remote link
[phrase@shortcut] » Shortcut
Note: [alias|any_of_above_links] » Custom link title

Tables:
	head1		head2	
colA1	colA2
colB1	colB2

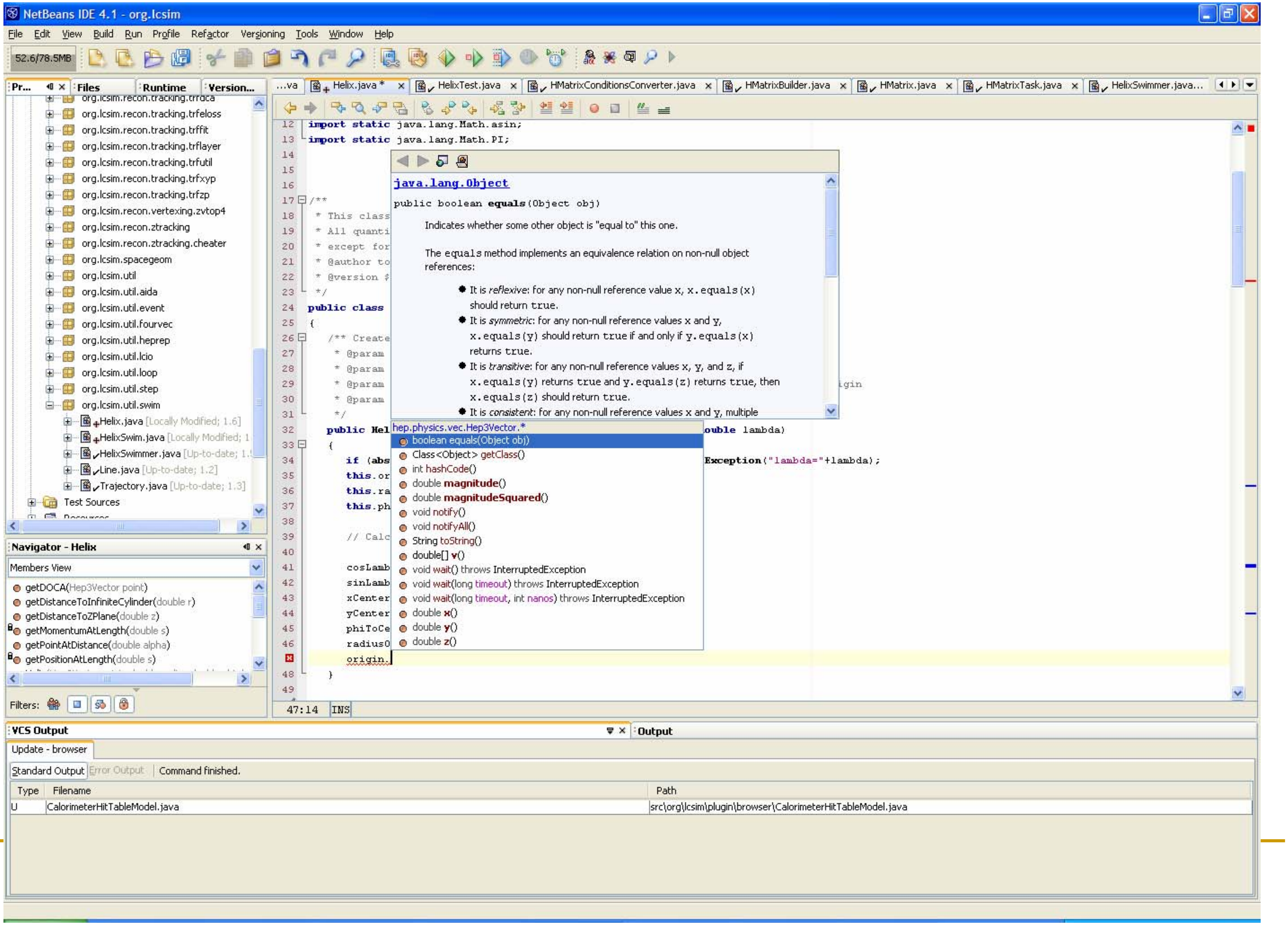
Details and full examples are in the full notation guide »

Powered by Atlassian Confluence, the Enterprise Wiki. (Version: 1.4.1 Build:#212 Jun 02, 2005) - Bug/feature request - Contact Administrators

How to contribute, continued

- Become a developer
  - All code in CVS
  - Use “Maven” as build system
    - Java project management tool
    - Single command
      - downloads dependencies, compiles code, runs tests, deploys code
  - “Contrib” area for in development example analysis, tools.
  - Most of reconstruction code is contributed by users
  - We strongly encourage developers to use IDE
    - Netbeans, Eclipse both free, easy to learn, very powerful
    - Use mevenide to teach IDEs about maven projects

Netbeans IDE



NetBeans IDE 4.1 - org.lcsim
File Edit View Build Run Profile Refactor Versioning Tools Window Help
52.6/78.5MB
Pr... Files Runtime Version...
...va Helix.java * HelixTest.java HMatrixConditionsConverter.java HMatrixBuilder.java HMatrix.java HMatrixTask.java HelixSwimmer.java...

- org.lcsim.recon.tracking.trfdca
- org.lcsim.recon.tracking.trfeloss
- org.lcsim.recon.tracking.trffit
- org.lcsim.recon.tracking.trflayer
- org.lcsim.recon.tracking.trfutil
- org.lcsim.recon.tracking.trfxyp
- org.lcsim.recon.tracking.trfzp
- org.lcsim.recon.vertexing.zvtop4
- org.lcsim.recon.ztracking
- org.lcsim.recon.ztracking.cheater
- org.lcsim.spacegeom
- org.lcsim.util
- org.lcsim.util.aida
- org.lcsim.util.event
- org.lcsim.util.fourvec
- org.lcsim.util.heprep
- org.lcsim.util.lcio
- org.lcsim.util.loop
- org.lcsim.util.step
- org.lcsim.util.swim
  - Helix.java [Locally Modified; 1.6]
  - HelixSwim.java [Locally Modified; 1
  - HelixSwimmer.java [Up-to-date; 1.
  - Line.java [Up-to-date; 1.2]
  - Trajectory.java [Up-to-date; 1.3]
- Test Sources
- Resources

Navigator - Helix

Members View

- getDOCA(Hep3Vector point)
- getDistanceToInfiniteCylinder(double r)
- getDistanceToZPlane(double z)
- getMomentumAtLength(double s)
- getPointAtDistance(double alpha)
- getPositionAtLength(double s)

Filters:

```
import static java.lang.Math.asin;
import static java.lang.Math.PI;


/**
 * This class
 * All quanti
 * except for
 * @author to
 * @version $
 */
public class
{
    /** Create
     * @param
     * @param
     * @param                                                  igin
     * @param
     */
    public Hel                               ouble lambda)
    {
        if (abs                              Exception("lambda="+lambda);
        this.or
        this.ra
        this.ph

        // Calc

        cosLamb
        sinLamb
        xCenter
        yCenter
        phiToCe
        radius0
        origin.
    }

```

java.lang.Object

`public boolean equals(Object obj)`

Indicates whether some other object is "equal to" this one.

The `equals` method implements an equivalence relation on non-null object references:

- It is *reflexive*: for any non-null reference value `x`, `x.equals(x)` should return `true`.
- It is *symmetric*: for any non-null reference values `x` and `y`, `x.equals(y)` should return `true` if and only if `y.equals(x)` returns `true`.
- It is *transitive*: for any non-null reference values `x`, `y`, and `z`, if `x.equals(y)` returns `true` and `y.equals(z)` returns `true`, then `x.equals(z)` should return `true`.
- It is *consistent*: for any non-null reference values `x` and `y`, multiple

hep.physics.vec.Hep3Vector.*

- boolean equals(Object obj)
- Class<Object> getClass()
- int hashCode()
- double **magnitude**()
- double **magnitudeSquared**()
- void notify()
- void notifyAll()
- String toString()
- double[] **v**()
- void wait() throws InterruptedException
- void wait(long timeout) throws InterruptedException
- void wait(long timeout, int nanos) throws InterruptedException
- double **x**()
- double **y**()
- double **z**()

47:14 INS

VCS Output | Output

Update - browser

Standard Output | Error Output | Command finished.

Type	Filename	Path
U	CalorimeterHitTableModel.java	src\org\lcsim\plugin\browser\CalorimeterHitTableModel.java

org.lcsim: Conclusions

- org.lcsim framework is developing rapidly
  - Support for older hep.lcd and LCIOPlugin will be phased out after this workshop
    - Most functionality already merged into org.lcsim framework
    - Will be working over next couple of months to port rest of code
- org.lcsim is a easy to get started with
  - Your contributions are welcome