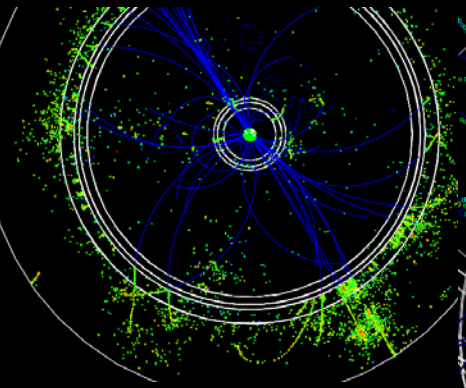
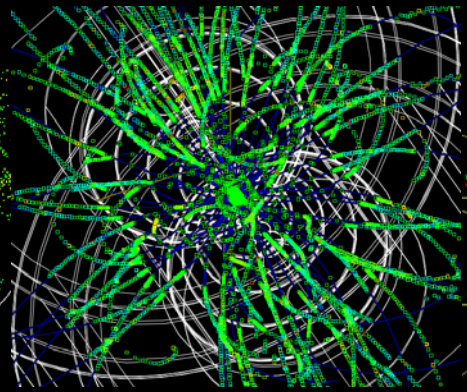
# ILC Detector Simulations: Overview of the US Framework

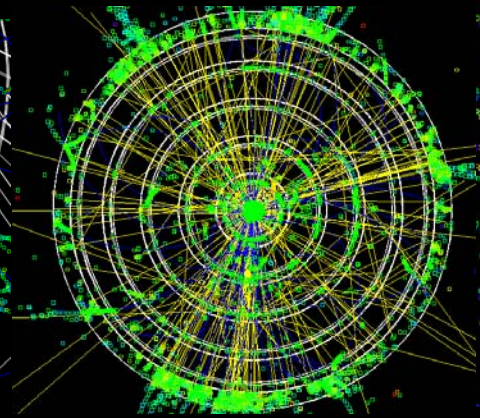
LDC: ttbar



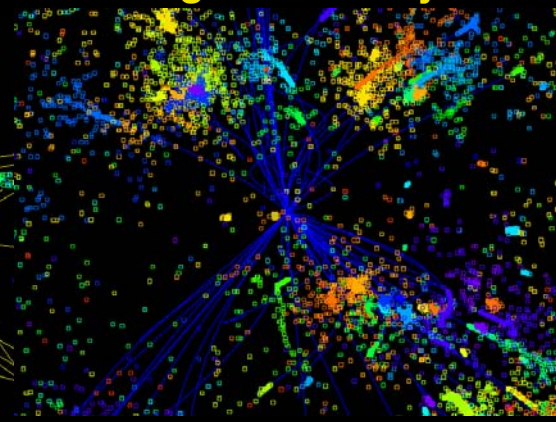
SiD May05: 100 muons



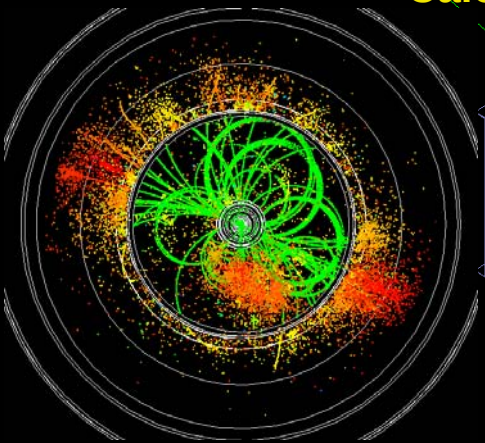
SiD Aug05: ttbar



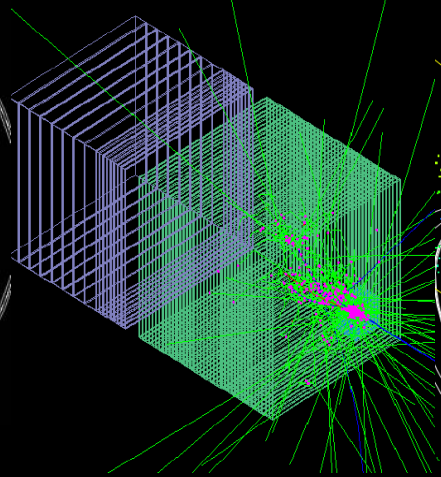
CDC Aug05: ttbar 6 jets



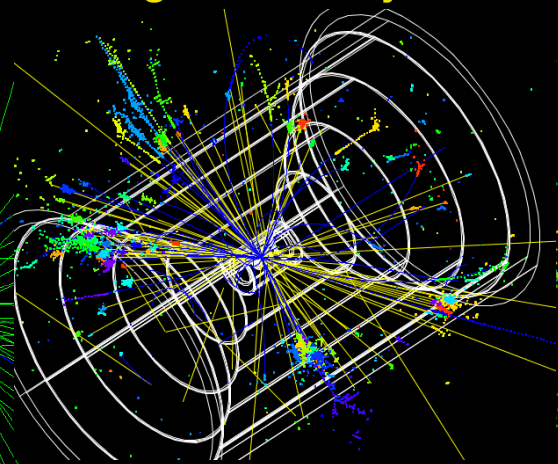
GLD: ttbar



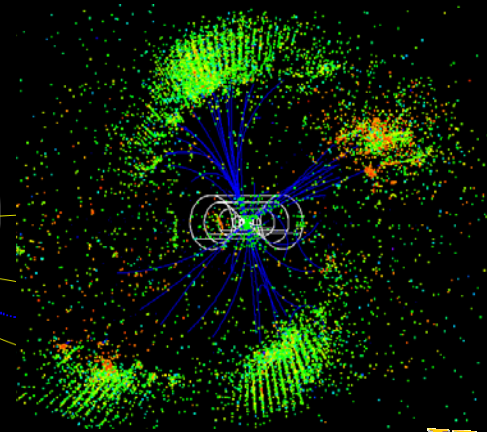
Calorimeter Testbeam



SiD Aug05: ttbar 6 jets



GLD: ZHiggs; $M_H$=120 GeV



Geant 4

World-Wide Studies of ILC Physics and Detectors

SLAC

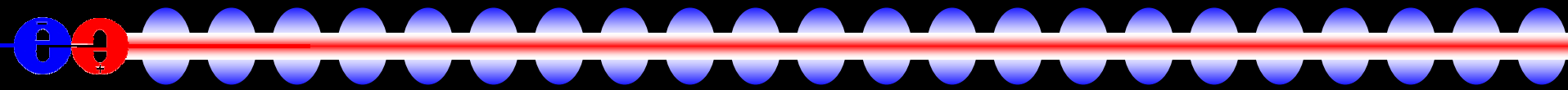
Tony Johnson -- SLAC
(stolen from Jeremy McCormick)
April 4 2006

SiD

FREE

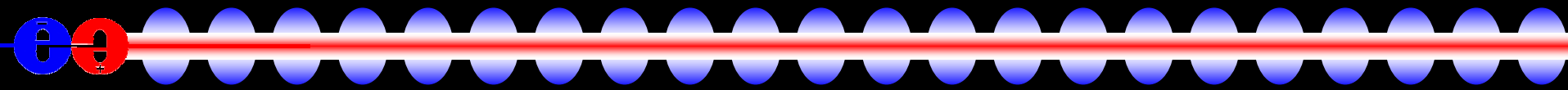
ECFA Study
Physics and Detectors for a Linear Collider

# Contents



- Goals
- Geometry Description
- SLIC – Simulation Package
- org.lcsim – Reconstruction Package
- Tools – JAS, WIRED, Grid
- Recent Developments
  - PFA, Tracking, WIRED enhancements

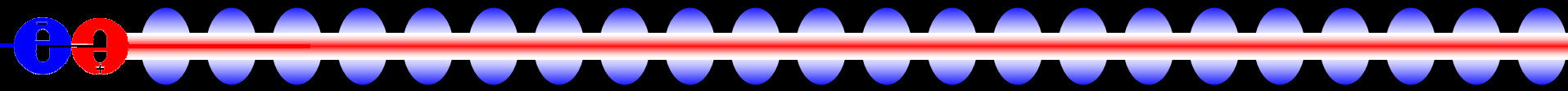
# Overview: Goals



- Facilitate contribution from physicists in different locations with various amounts of time available
- Use standard data formats, when possible.
- Provide a general-purpose framework for physics software development.
- Develop a suite of reconstruction and analysis algorithms and sample codes.
- Simulate benchmark physics processes on different full detector designs.

# Overview: Key Features

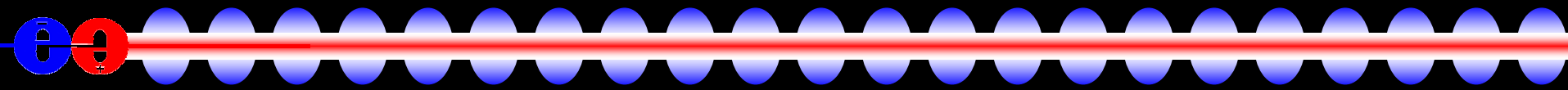


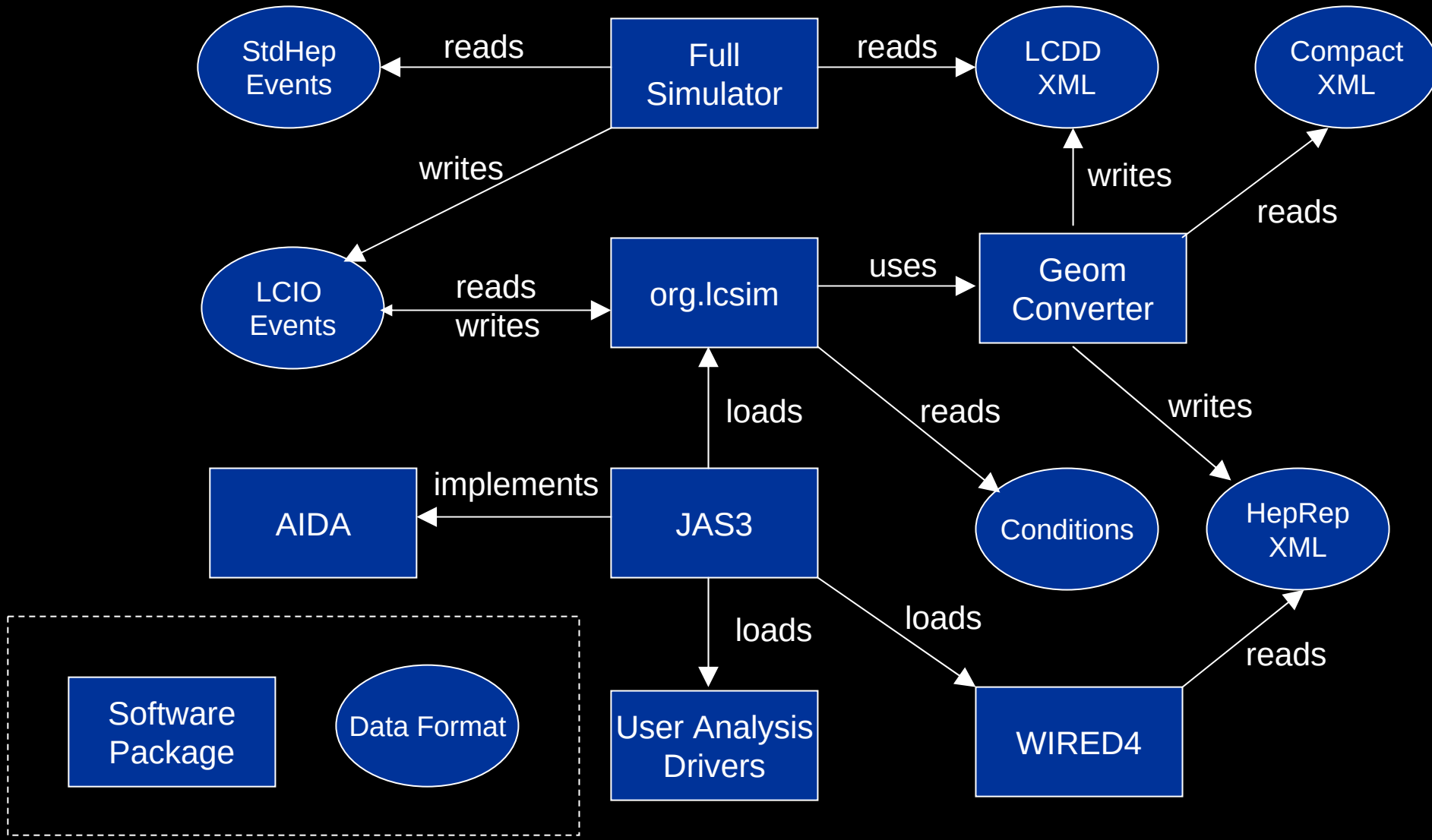


- Cross-platform compatibility
  - Java reconstruction software is write/run anywhere.
    - Maven for easy builds
    - Encourage use of full IDE's for development
      - Netbeans, Eclipse
  - The simulation software runs on OSX, Linux and Windows
    - GNU Autoconf/Make build system
- Supports ILC software standards
  - AIDA, LCIO, StdHep, HepRep
- Easy to model different detector designs
  - Geometry, materials, readout, and IDs easily customized.
  - Write hooks to Java classes for detector components
  - Convert to several different output geometry formats

# Overview: Framework Diagram

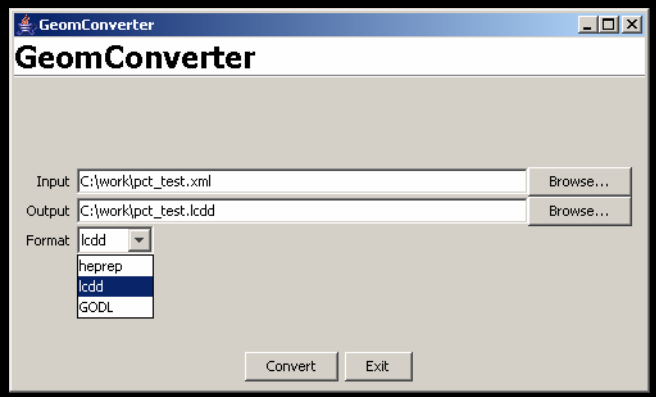






# Detectors: Geometry Converter





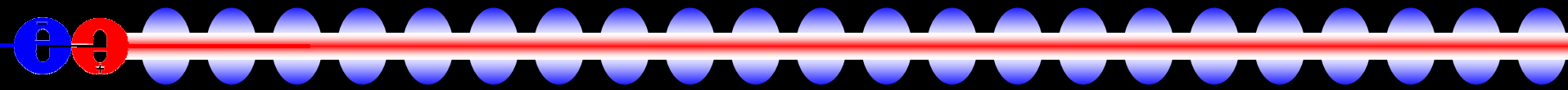




- Convert from compact to
  - LCDD for simulator (SLIC)
  - HepRep for event display (WIRED4)
  - GODL for fast MC (Lelaps)
  - Java Objects for reconstruction (org.lcsim)

# Detectors: Zip Files





- Detector conditions bundled into downloadable Zip file
- Can also point org.lcsim to custom and development versions
- Canonical detectors location
  - http://www.lcsim.org/detectors
- sid00 examples
  - Zip file
    - http://www.lcsim.org/detectors/sid00.zip
  - LCDD
    - http://www.lcsim.org/detectors/sid00/sid00.lcdd
  - Compact
    - http://www.lcsim.org/detectors/sid00/compact.xml

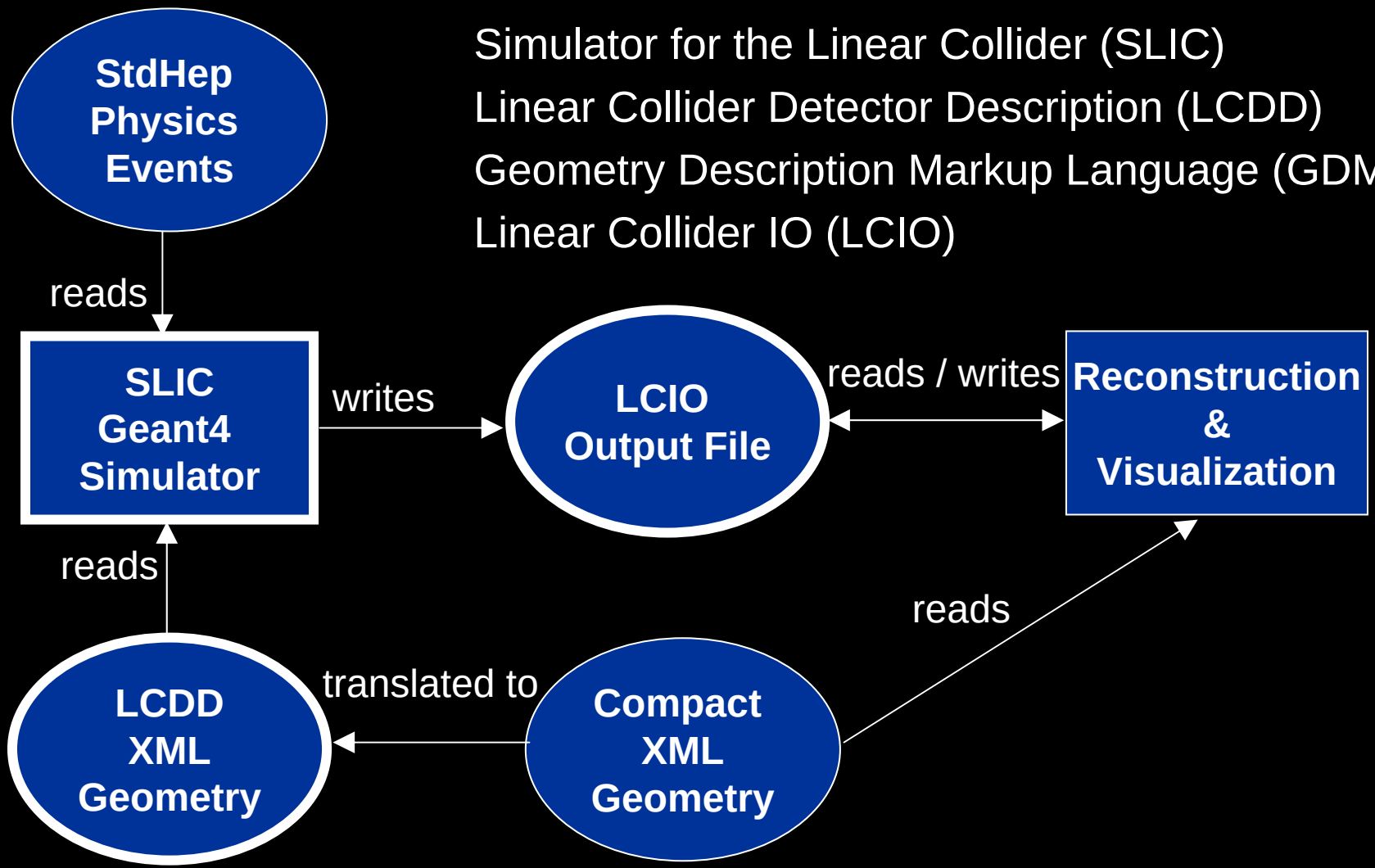
sid00

```
|-- ClusterParameters.properties
|-- IDEfficiency.properties
|-- IndividualParticleReconstruction.properties
|-- LongitudinalHMatrix.hmx
|-- SamplingFractions
|   |-- EMBarrel.properties
|   |-- EMEndcap.properties
|   |-- ForwardEMEndcap.properties
|   |-- ForwardLuminosityMonitor.properties
|   |-- HADBarrel.properties
|   |-- HADEndcap.properties
|   |-- LumEndcap.properties
|   |-- MuonBarrel.properties
|   `-- MuonEndcap.properties
|-- SimpleTrack.properties
|-- TrackParameters
|   |-- FullBarrelResolutionBc.ini
|   |-- FullBarrelResolutionNbc.ini
|   `-- sid00_lcdtrk.input
|-- TrackParameters.properties
|-- compact.xml
|-- detector.properties
|-- digisim
|   `-- digisim.steer
|-- sid00.lcdd
`-- sid00.zip
```

# Simulation: SLIC

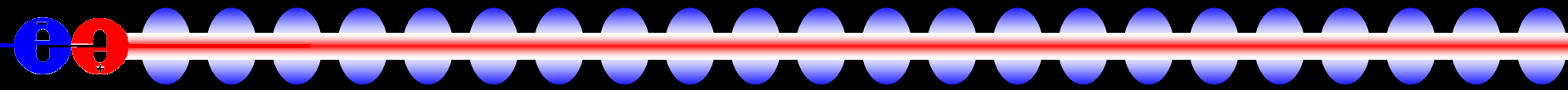


Simulator for the Linear Collider (SLIC)
Linear Collider Detector Description (LCDD)
Geometry Description Markup Language (GDML)
Linear Collider IO (LCIO)



# Simulation: SLIC Commands





- All command-line options have equivalent Geant4 command
- Sample command

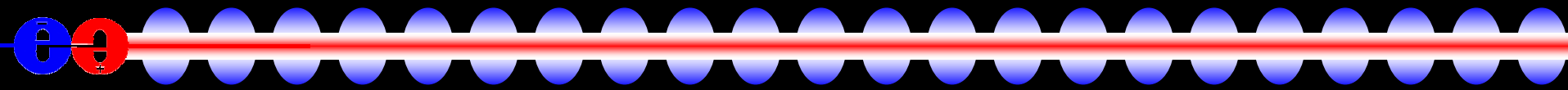
```
slic -g geometry.lcdd -i events.stdhep -x -O -l LCPhys -r 1000
```

- Equivalent macro

```
/lcdd/url geometry.lcdd
/run/initialize
/physics/select LCPhys
/generator/filename events.stdhep
/lcio/fileExists delete
/lcio/autoname
/run/beamOn 1000
```

# Simulation: Detector Description





## LCDD

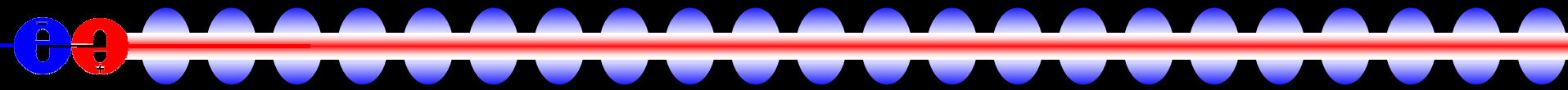
- Identifiers
- Sensitive Detectors
- Regions
- Physics Limits
- Visualization
- Magnetic Fields

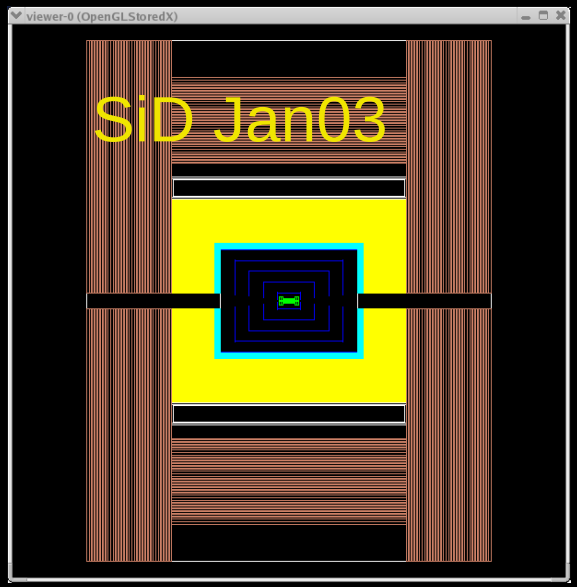
## GDML

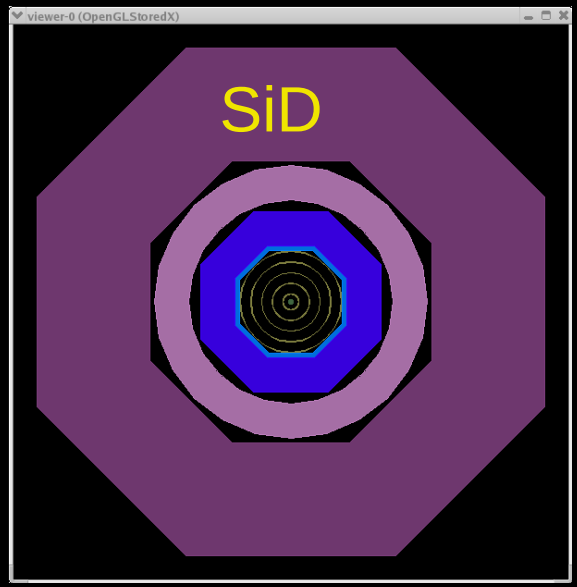
- Expressions (CLHEP)
- Materials
- Solids
- Volumes

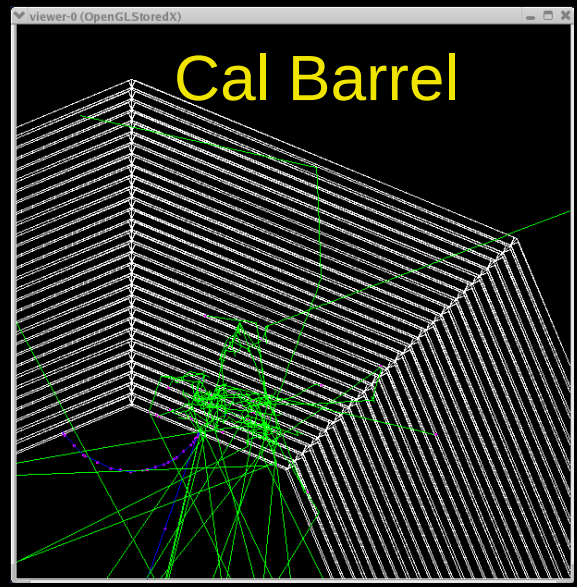
# Simulation: Example Geometries

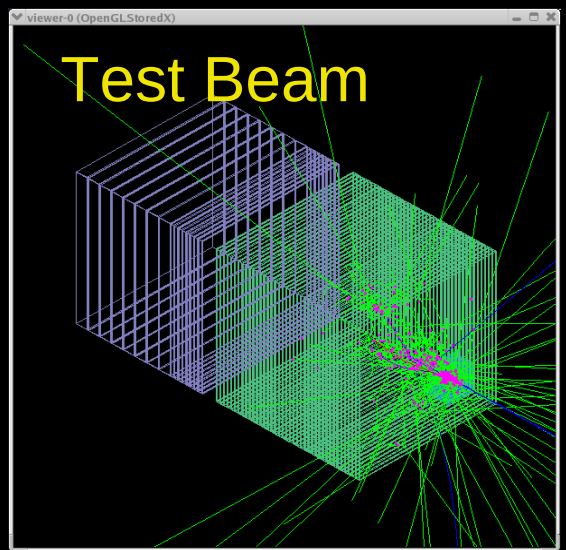




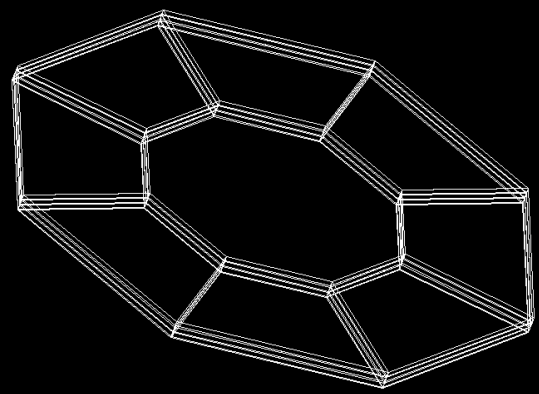


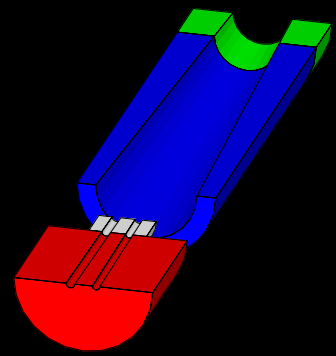




Cal Endcap



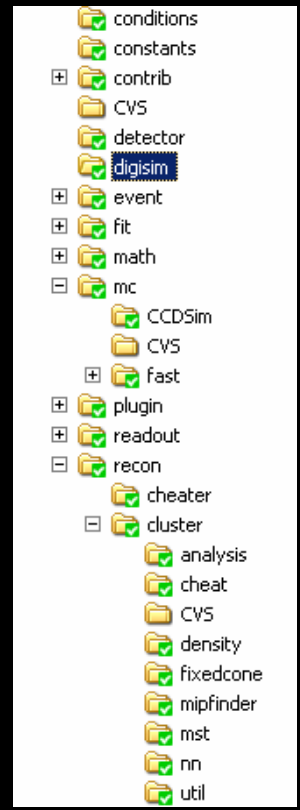
MDI-BDS



# Reconstruction and Analysis Software



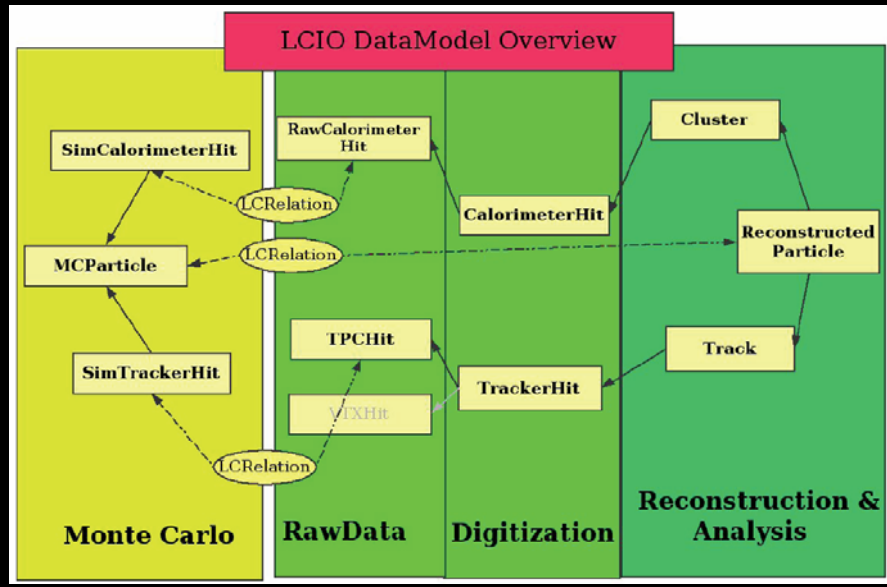
- org.lcsim
  - Reconstruction and analysis package
- GeomConverter
  - Geometry system
- FreeHep
  - Physics and graphics utilities
- JAS3
  - Studio application for analysis and development
- WIRED4
  - Event display
- LCIO
  - Object model and persistency
- AIDA
  - Plotting API and data format

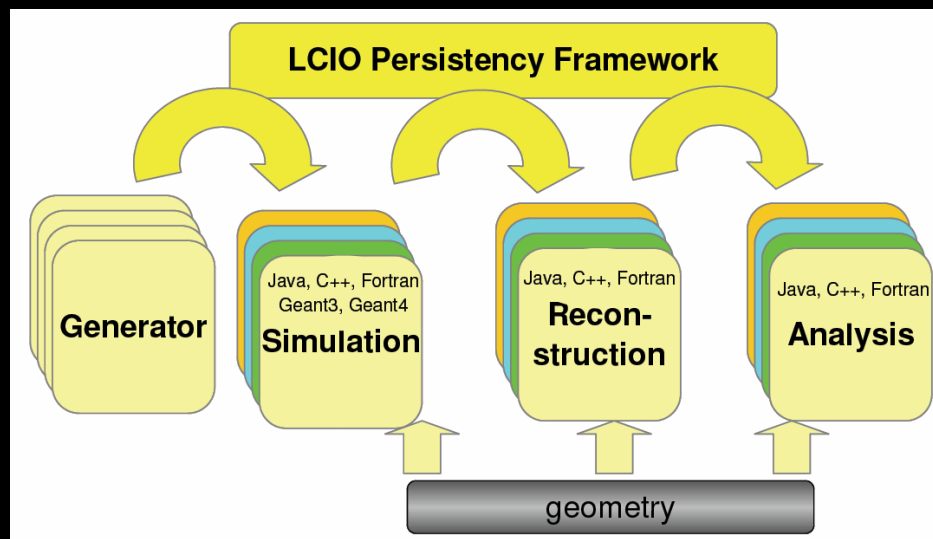


# LCIO



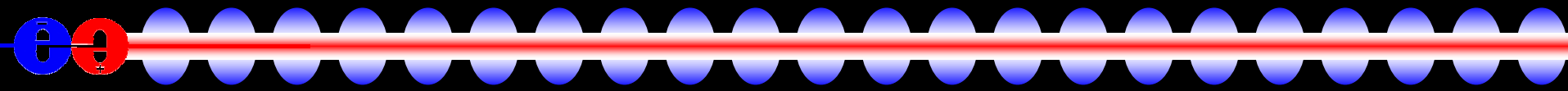
- Object model and persistency
  - Events
    - Monte Carlo
    - Raw
    - Event and run metadata
  - Reconstruction
  - Parameters, relations, attributes, arrays, generic objects, …
- All the ILC simulators write LCIO
  - Enables cross-checks between data from different simulators
  - Read/write LCIO from
    - Fast MC / Full Simulation
    - Different detectors
    - Different reconstruction tools



# LCIO: Data Samples





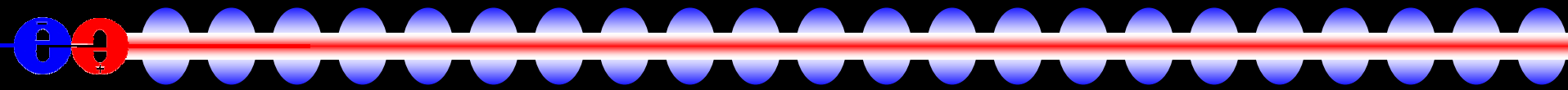
- LCIO data samples available via anonymous FTP
  - http://www.lcsim.org/datasets/ftp.html
- Data sets
  - ILC500
    - 500 GeV machine parameters
  - ILC1000
    - 1 TeV machine parameters
  - singleParticle
    - Single particle diagnostic events
  - Zpole
    - Zpole diagnostic events

Organization

- **[event type]** - complex or single particle event type, e.g. ZZ, ZPole, muons, etc.
  - **stdhep** - input StdHep files used to generate the events
  - **[detector name]** - detector geometry tag, such as sidaug05
    - **[data file format]** - output datafile format, e.g. LCIO or SIO
      - **[simulator]** - simulator that generated the events, e.g. lcdg4, slic, lelaps, mokka, etc.
        - **logs** - simulator job logs

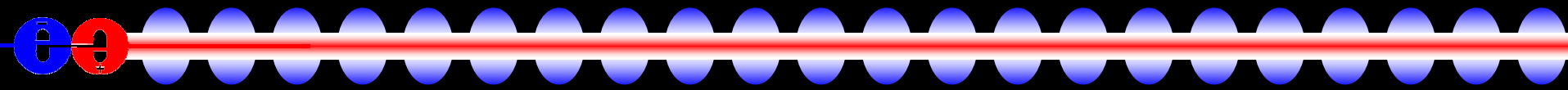
# org.lcsim: Goals





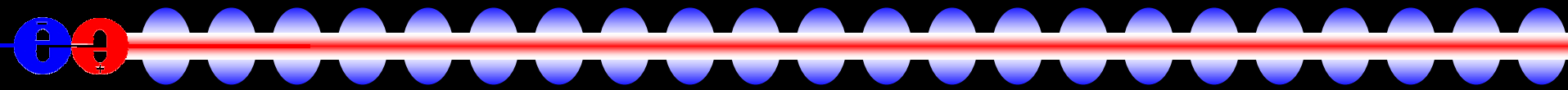
- Retain core functionality from hep.lcd package
  - Full suite of reconstruction and analysis tools
- Update to use latest LCIO for IO and as basis for simulation, raw data and reconstruction event formats
- Insulate users from raw LCIO structures
- Update and simplify framework using experience from hep.lcd
  - Provide good tutorial documentation
- Detector Independence
  - 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

# Reconstruction/Analysis Overview



- Java based reconstruction and analysis package
  - Runs standalone or inside Java Analysis Studio (JAS)
  - Detector Geometry Independent
    - Read properties of detectors at runtime
    - Encourage algorithm developers to avoid detector and geometry assumptions
  - Fast MC $\rightarrow$ Smeared tracks and calorimeter clusters
  - Full Event Reconstruction
    - detector readout digitization (CCD pixels & Si $\mu$-strips)
    - *ab initio* track finding and fitting for ~arbitrary geometries
    - multiple calorimeter clustering algorithms
    - Individual Particle reconstruction (cluster-track association)
  - Analysis Tools (including WIRED event display)
  - Physics Tools (Vertex Finding, Jet Finding, Flavor Tagging)
  - Beam background overlays at detector hit level
  - Derived from earlier hep.lcd package
    - Updated to use Java 1.5, LCIO
    - More detailed reconstruction algorithms

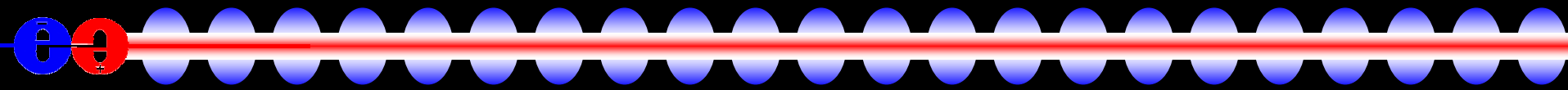
# org.lcsim: Status



- **Physics Utilities**
  - Stdhep Reader
  - 3, 4-vector utilities
  - diagnostic generator
  - Jet finder, event shape utilities
- **Conditions framework**
  - 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**
  - All reconstruction algorithms and user analysis written as Drivers
  - Read/Write access to event
  - Drivers can be chained and nested
- **FastMC**
  - parameterized track and cluster smearing
  - Smearing constants read from conditions system
  - Produces ReconstructedParticles
- **Digitization**
  - Digisim–Calorimeter digitization
  - *Tracker, Vertex digitization*
- **Clustering**
  - Cheater
  - Cone
  - Nearest Neighbor
  - *Minimal Spanning Tree*
- **Tracking**
  - SLDWeightMatrix, Kalman Filter, TRF
  - Track Cheater
- **Vertex Fitting**
  - ZVTop4
- **Analysis Examples**
  - Cluster Diagnostics
  - SLICDiagnostics
  - Particle Flow Analysis
  - ClusterID
  - ParticleID

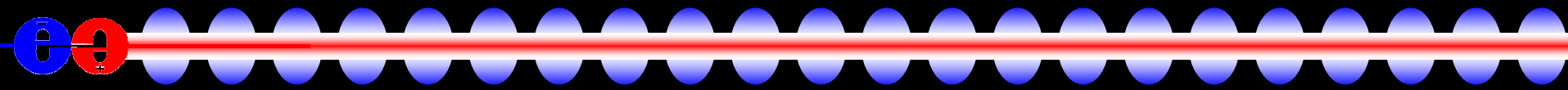
# Reconstruction and Analysis





- Org.lcsim contains a “contrib” area
- Encourage users to place ongoing work in this area, even if still “in progress”
  - Last six months have seen greatly increased use of this area
    - ClusterAnalysis utilities
    - “SODTracker”
    - “Garfield Track Finder” (K0 decays)
    - “PFA Template”
    - …

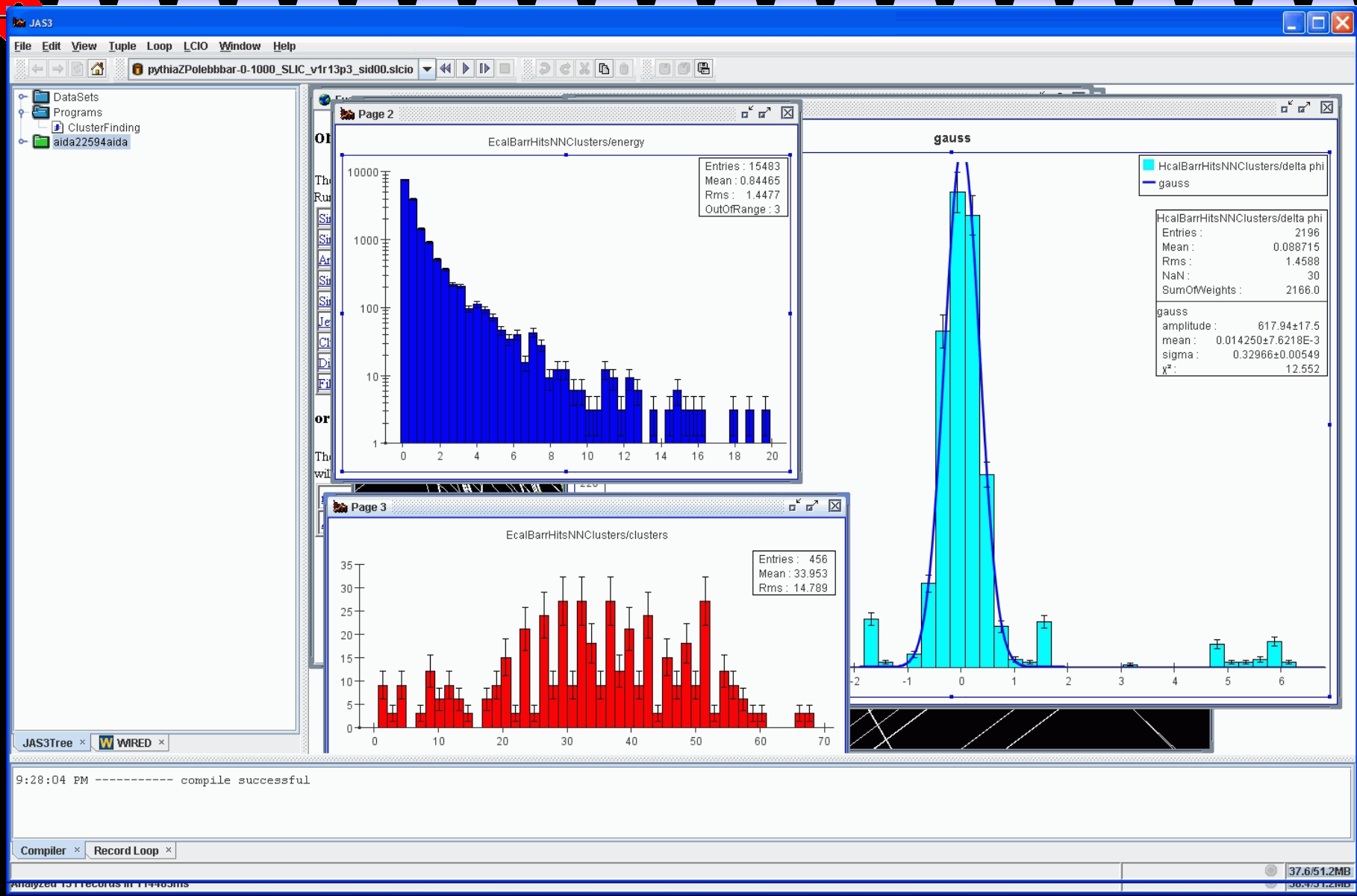
# Related Talks



- PFA, PFA “Template”, DigiSim
  - Steve Magill, Mat Charles, Norm Graf
- Tracking
  - Norm Graf
- Demo/Tutorials on Wednesday

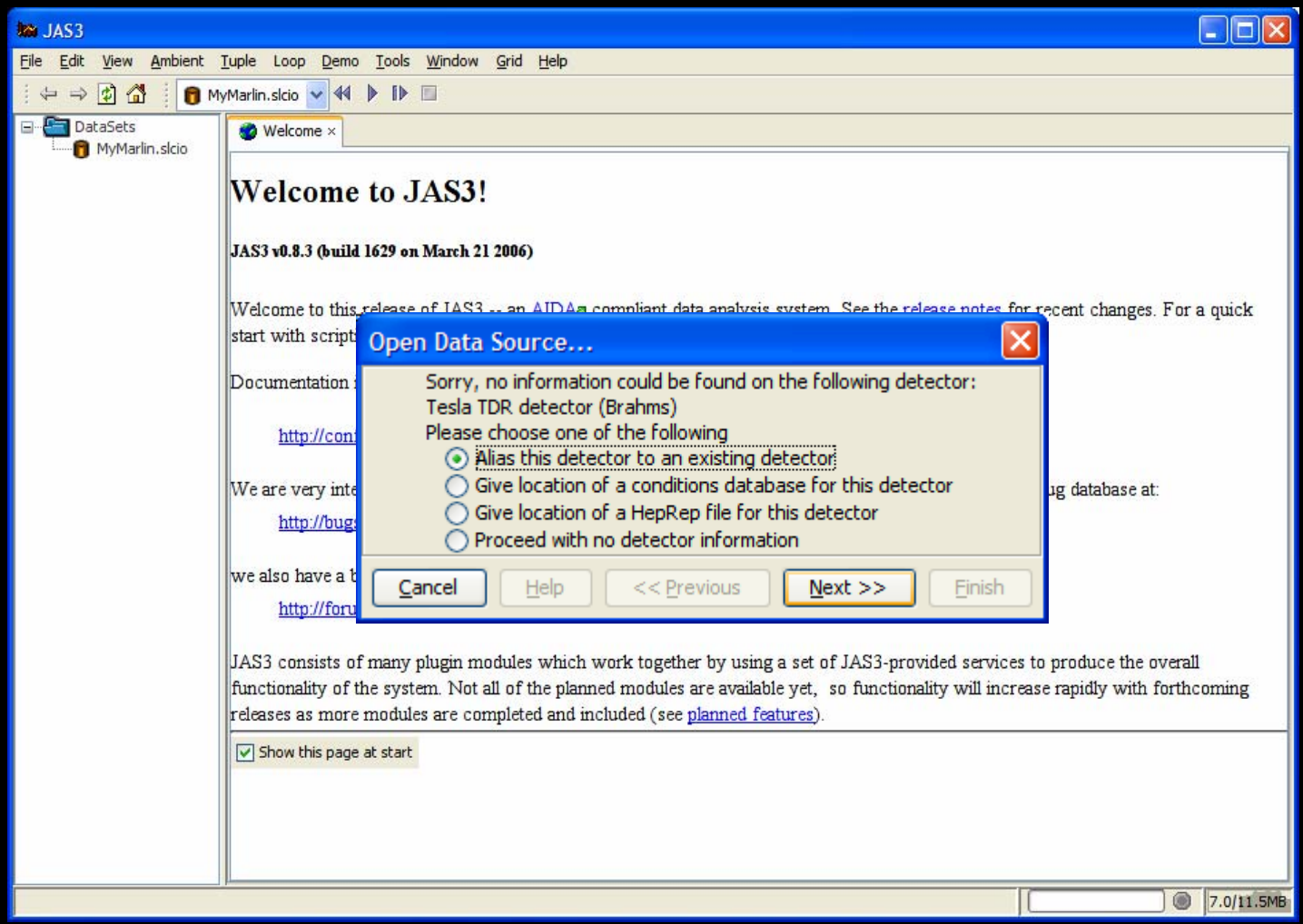
# Tools: JAS3




JAS3
File Edit View Tuple Loop LCIO Window Help
pythiaZPolebbbar-0-1000_SLIC_v1r13p3_sid00.slcio
DataSets
Programs
ClusterFinding
aida22594aida
JAS3Tree
WIRED
Page 2
EcalBarrHitsNNClusters/energy
Entries : 15483
Mean : 0.84465
Rms : 1.4477
OutOfRange : 3
Page 3
EcalBarrHitsNNClusters/clusters
Entries : 456
Mean : 33.953
Rms : 14.789
gauss
HcalBarrHitsNNClusters/delta phi
gauss
HcalBarrHitsNNClusters/delta phi
Entries : 2196
Mean : 0.088715
Rms : 1.4588
NaN : 30
SumOfWeights : 2166.0
gauss
amplitude : 617.94±17.5
mean : 0.014250±7.6218E-3
sigma : 0.32966±0.00549
χ² : 12.552
9:28:04 PM ---------- compile successful
Compiler
Record Loop
37.6/51.2MB

# Recent Developments: Interoperability



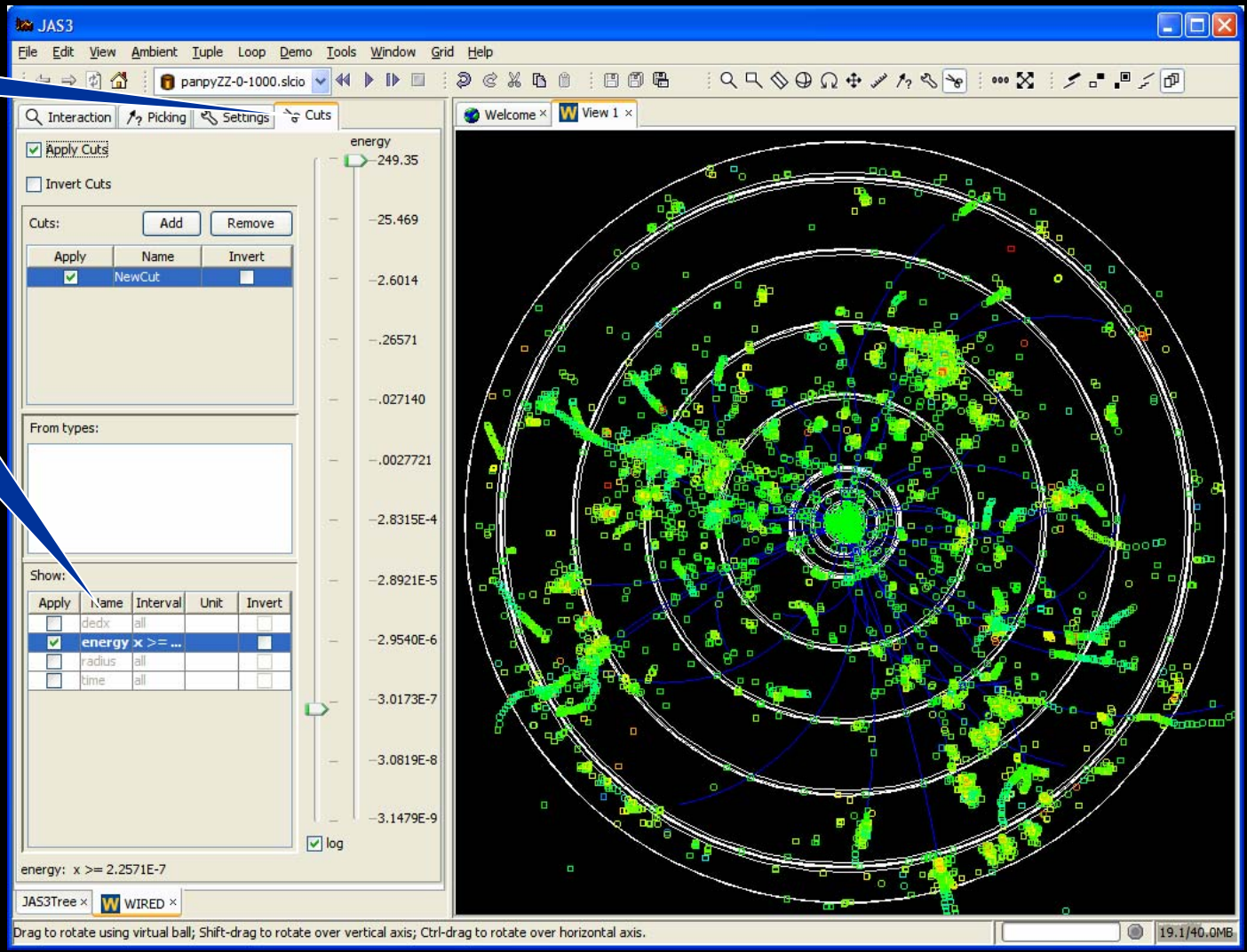


# Recent Developments: WIRED



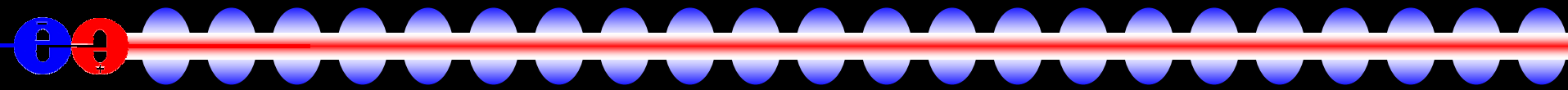
New Cuts Tab

Cuts can be defined interactively and persist across events and sessions

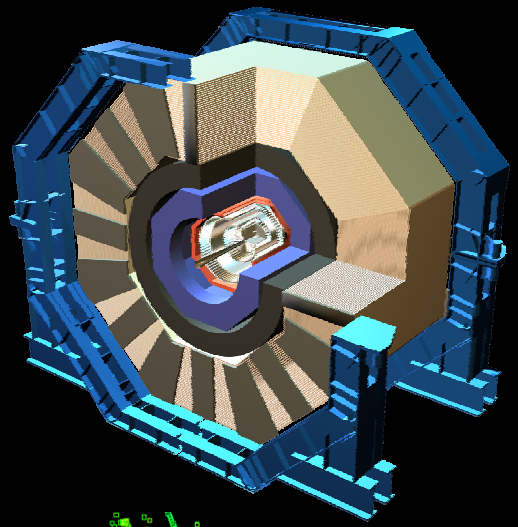


# Interoperability: Event Display

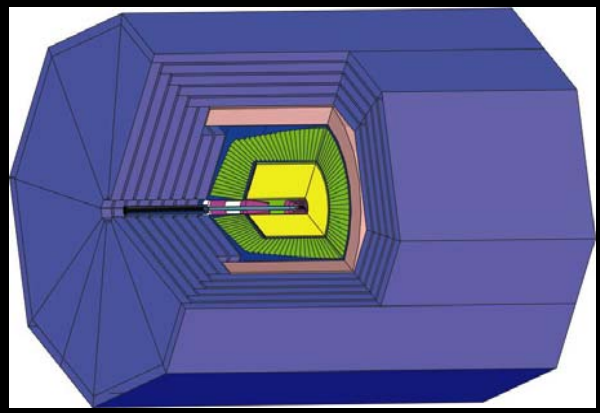




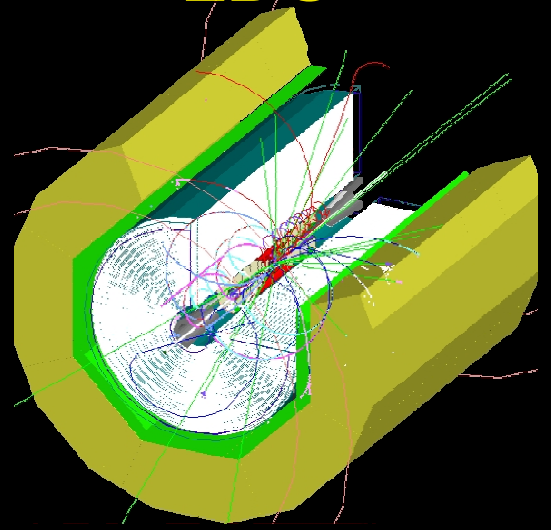
SiD

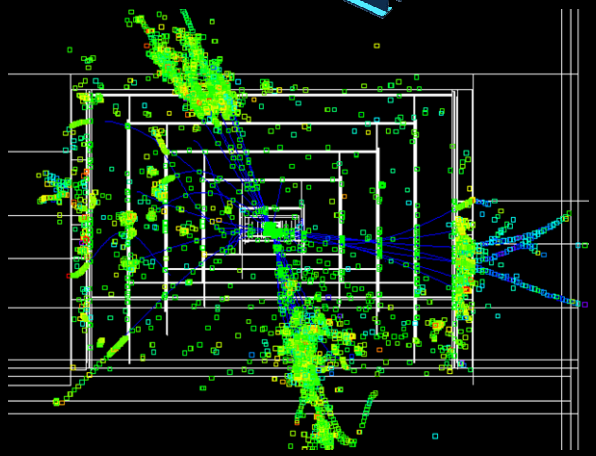


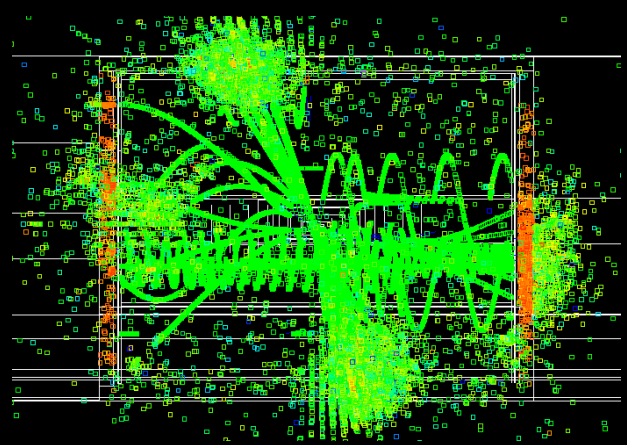
GLD

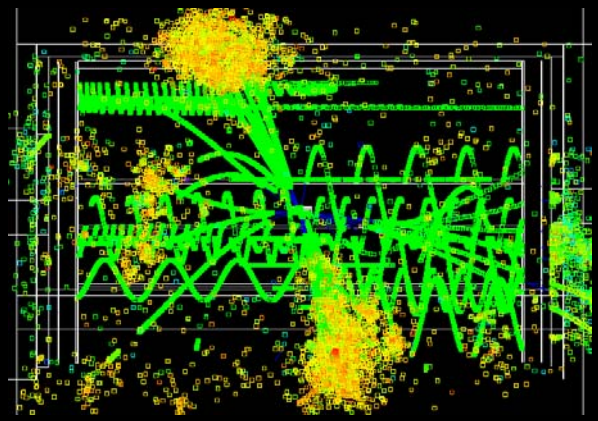


LDC





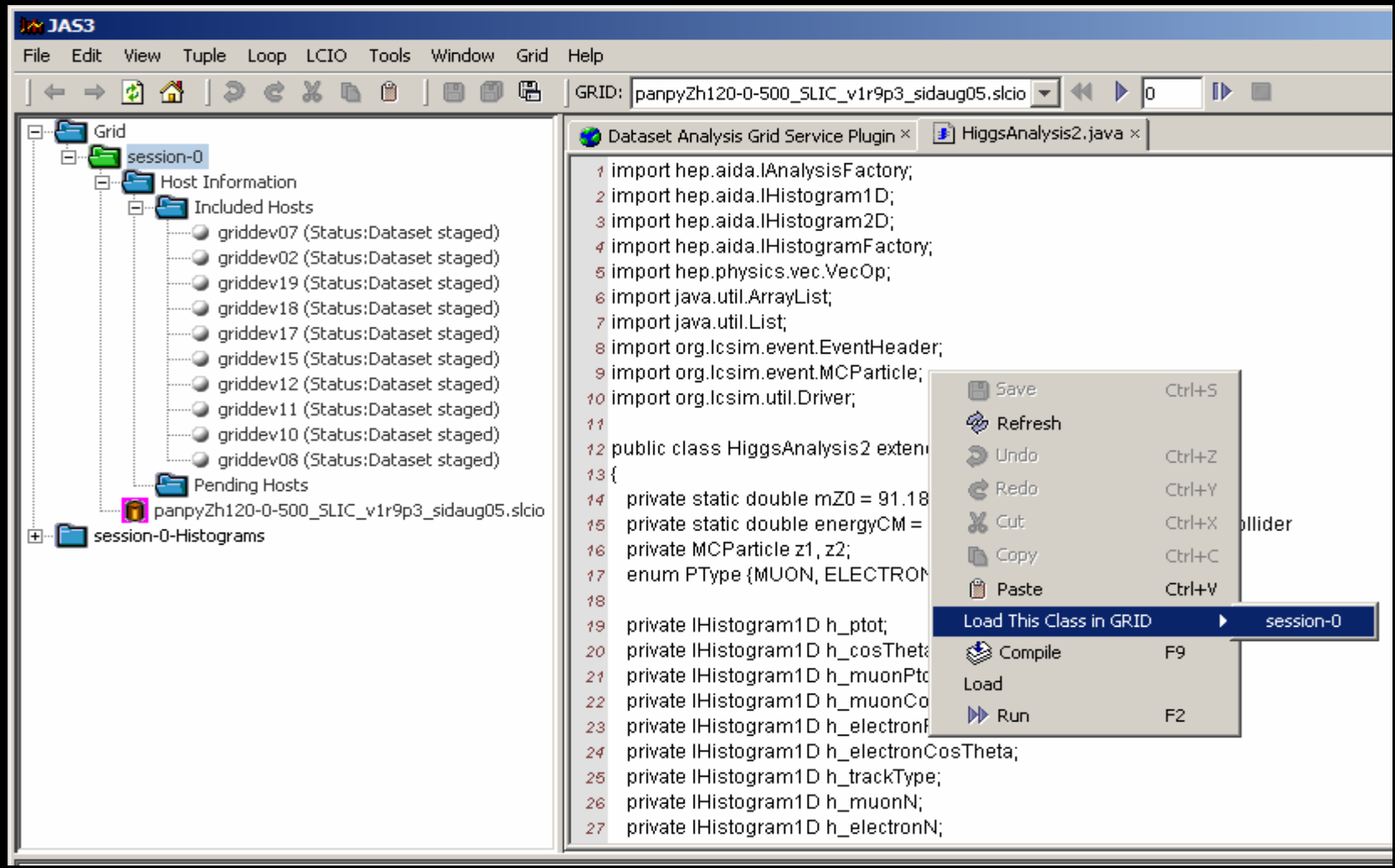




Z Higgs ($M_H$=120 GeV) → same simulator, three different full detector geometries

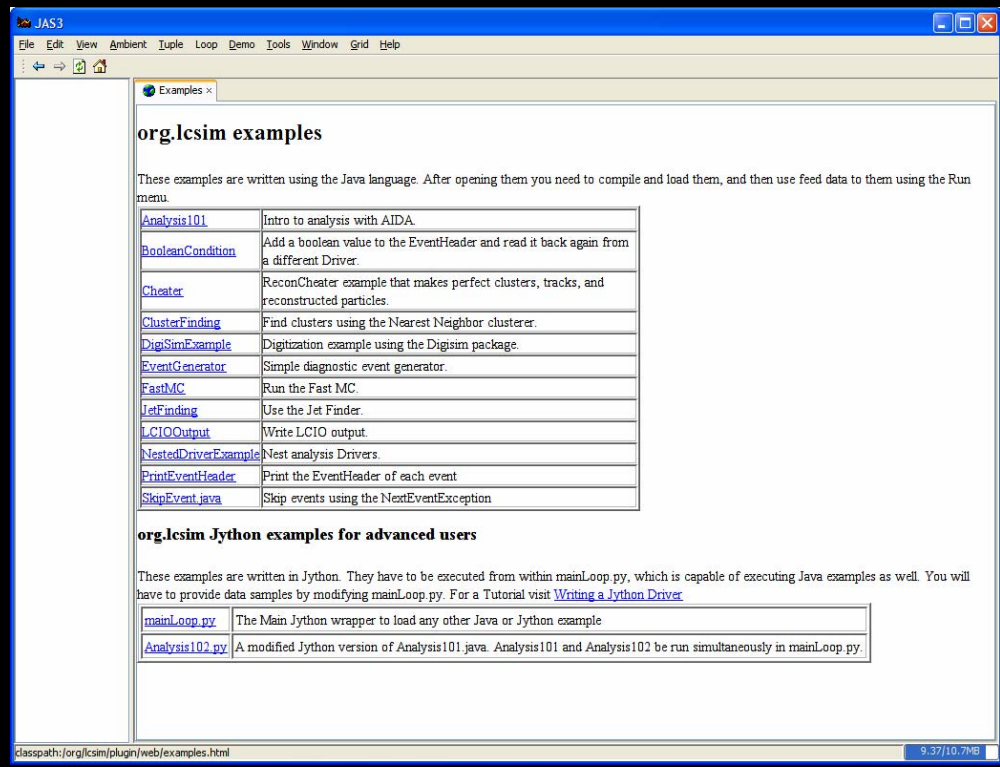
# Recent Developments: Grid Based Analysis





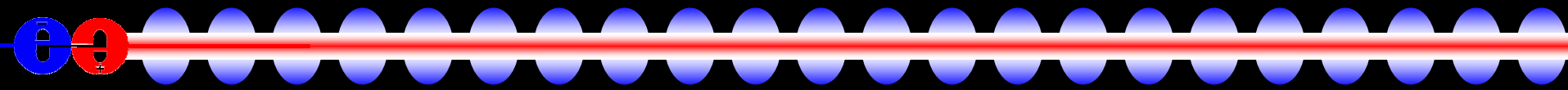
# Getting Started

- Anyone is welcome to use/contribute to this framework
- Easiest way to start
  - http://lcsim.org
  - Follow “Getting Started Link”
  - After installing JAS use the “Help, Examples”



# Documentation

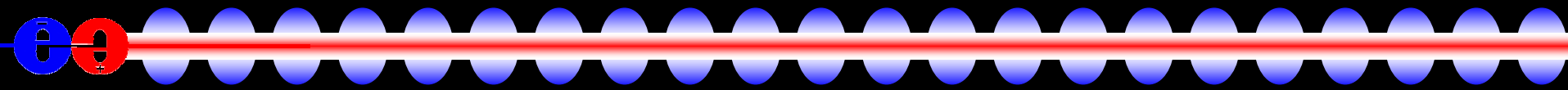




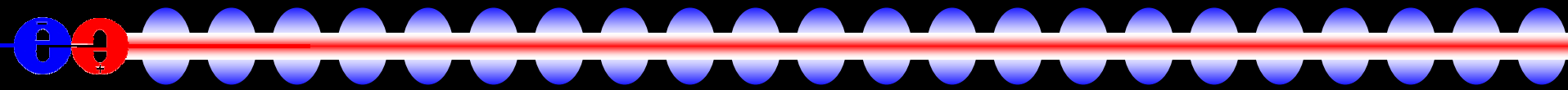
- IIC Wiki (confluence) used for supplementary docs
  - HTML-like syntax
  - Export to PDF
  - Need account to contribute (email Tony Johnson)
- Doxygen for C++ documentation
  - SLIC, LCDD
- JavaDoc for Java documentation
  - org.lcsim, GeomConverter
  - Generated automatically along with website by Maven

# Links





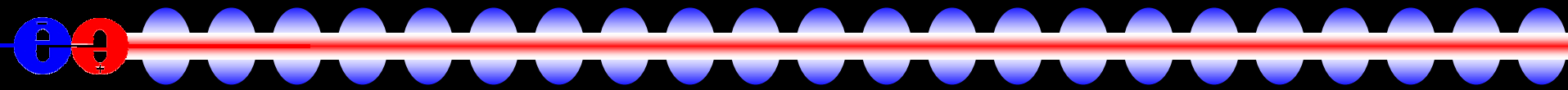
- Wiki - http://confluence.slac.stanford.edu/display/ilc/Home
- lcsim.org - http://www.lcsim.org
- org.lcsim - http://www.lcsim.org/software/lcsim
- Software Index - http://www.lcsim.org/software
- Detectors - http://www.lcsim.org/detectors
- ILC Forum - http://forum.linearcollider.org
- LCIO - http://lcio.desy.de
- SLIC - http://www.lcsim.org/software/slic
- LCDD - http://www.lcsim.org/software/lcdd
- JAS3 - http://jas.freehep.org/jas3
- AIDA - http://aida.freehep.org
- WIRED - http://wired.freehep.org



# Extra Slides

# Overview: Key Features (cont'd)

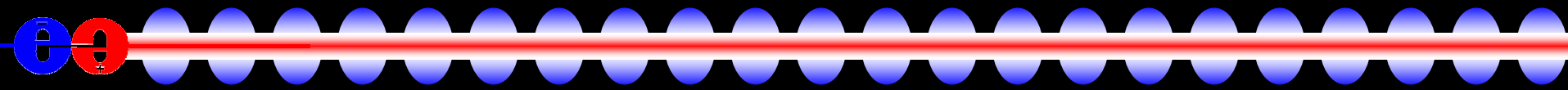




- Flexibility
  - Reconstruction and analysis
    - JAS3 analysis environment
      - Load/unload Java classes interactively
    - Java libraries automatically downloaded
    - FreeHep codebase
      - AIDA, Wired, HepRep, ROOT, StdHep, …
  - Simulation
    - XML detector input → No user C++ code required.
    - Drive from command-line or macros (1-to-1)
    - Geant4 MC toolkit
      - Multiple physics list selection

# Detectors: Compact Detector Description

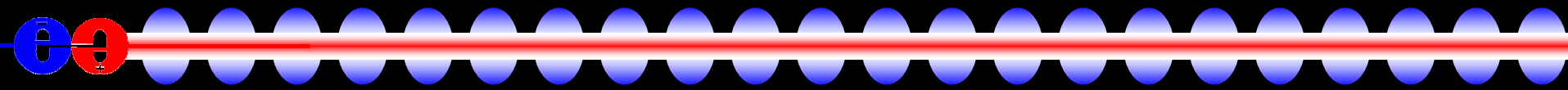




- Shorthand format for detector description
  - SiD 00 → 600 lines of XML
- Describes
  - Detector metadata
  - Materials
  - Readouts and identifiers
  - Detector components
    - Dimensions
    - Layering
  - Magnetic field

# Detectors: Compact XML Example

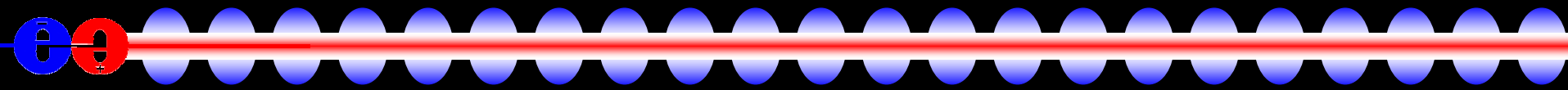




## Two layer stacks in an ECAL barrel

```
 <detector id="2" name="EMBarrel" type="CylindricalBarrelCalorimeter"
readout="EcalBarrHits">
      <dimensions inner_r = "150.1*cm" outer_z = "208.0*cm" />
      <layer repeat="20">
        <slice material = "Tungsten" thickness = "0.25*cm" />
        <slice material = "G10" thickness = "0.068*cm" />
        <slice material = "Silicon" thickness = "0.032*cm" sensitive = "yes" />
        <slice material = "Air" thickness = "0.025*cm" />
      </layer>
      <layer repeat="10">
        <slice material = "Tungsten" thickness = "0.50*cm" />
        <slice material = "G10" thickness = "0.068*cm" />
        <slice material = "Silicon" thickness = "0.032*cm" sensitive = "yes" />
        <slice material = "Air" thickness = "0.025*cm" />
      </layer>
  </detector>
```

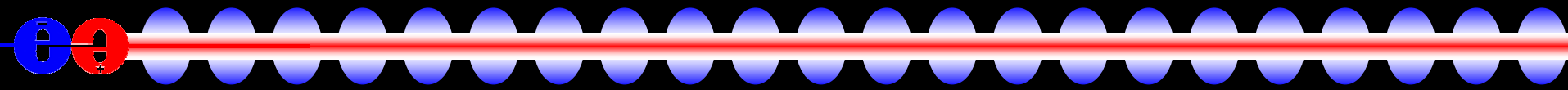
# Detectors: Repository



- Detector descriptions stored in CVS Project *LCDetectors*
- Easy to add new detectors based on existing ones
- All models in CVS periodically replicated to WWW locations
- Models
  - cdcaug05, cdcaug05_ecal150, cdcaug05_np, cdcaug05_rpchcal, gldaug05, hd3_1-oct05, ld3_1-oct05, ldcaug05, sdfeb05, sdjan03, sid00, sidaug05, sidaug05_20mr, sidaug05_4tesla, sidaug05_gemhcal, sidaug05_np, sidaug05_polyhedra, sidaug05_scinthcal, sidaug05_tcmt, sidmay05, sidmay05_20mr, sidmay05_2mr, sidmay05_np, sidmay05_scinthcal

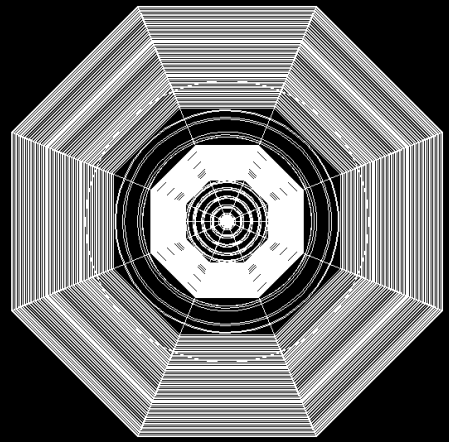
# Detectors: Geometry Displays



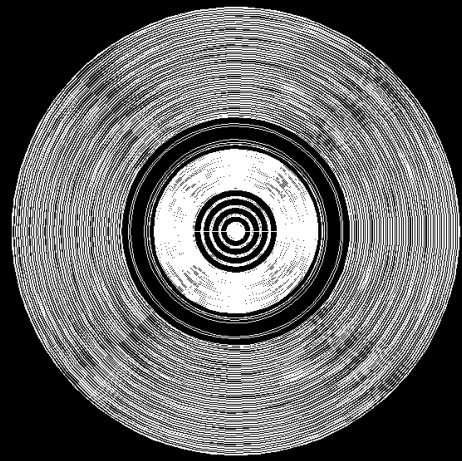


cdcaug05_ecal150

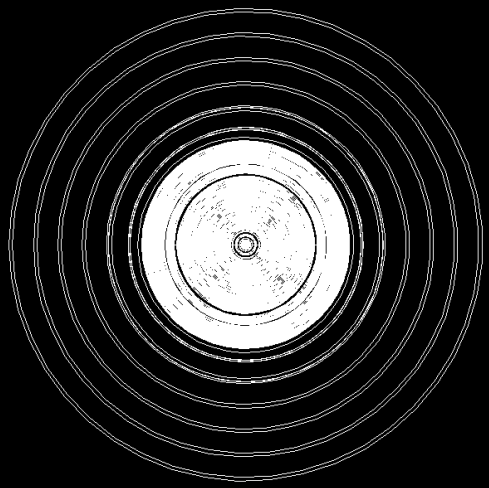
sidaug05_polyhedra



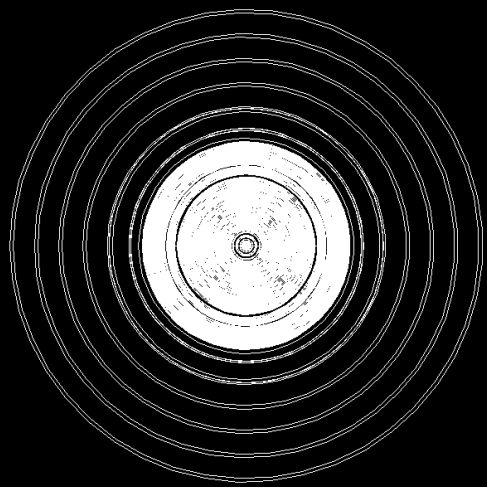
sid00



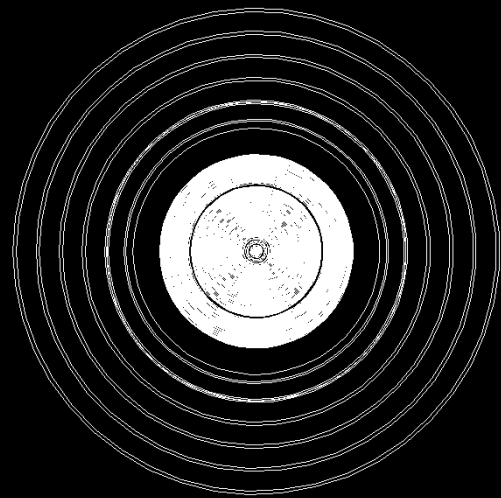
hd3_1-oct-05



ld3_1-oct-05



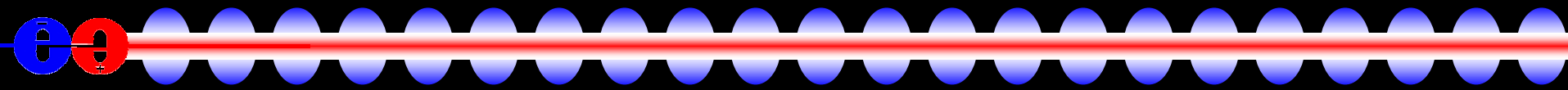
gldaug05



Geant4-generated HepReps drawn by WIRED4.

# Simulation: LCDD Data Binding

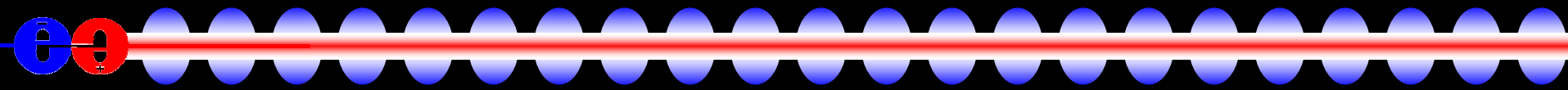




| Area | Root Element | Geant4 Class(es) |
|---|---|---|
| Sensitive Detectors | <sensitive_detectors> | G4VSensitiveDetector |
| Identifiers | <iddict> | NA (custom classes) |
| Regions | <regions> | G4Region, G4VUserRegionInformation |
| Physics Limits | <limits> | G4UserLimits |
| Visualization | <display> | G4VisAttributes |
| Magnetic Fields | <fields> | G4MagneticField |
| Constants | <define> | NA (CLHEP expressions) |
| Materials | <materials> | G4Material, G4Element |
| Shapes | <solids> | G4VSolid |
| Volumes | <structure> | G4LogicalVolume, G4VPhysicalVolume |

# Event Display: WIRED and HepRep

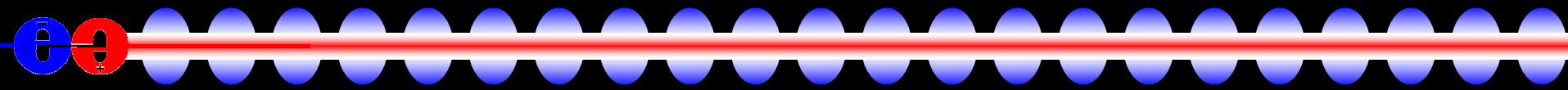




- HepRep file format
  - Generic format for event display
  - GeomConverter streams HepRep data to WIRED.
  - Also written out by Geant4
- WIRED4
  - Layers for detector, hits, MCParticles, etc.
  - Rotation, zoom, panning, picking
  - Interactive activation/deactivation of objects in display
  - Writes to PNG, JPG, WMF, HepRep, etc.

# AIDA and Plotting

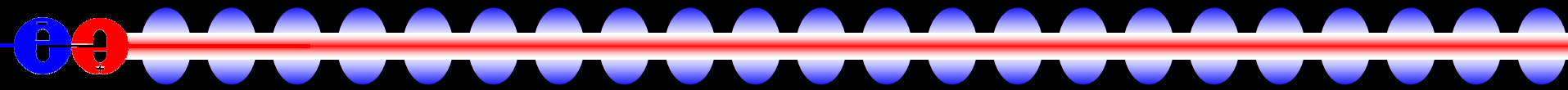




- JAS3
  - Interactive plotting
  - Supported plot types
    - Histograms, clouds, profiles (1D + 2D)
- AIDA API supports 3D plot creation
  - Open Scientist, PAIDA can plot these.
- N-tuples still not very functional, but doesn't seem to be too limiting to analysis. (?)
  - Need binary format → HDF5 ?
- Implementations
  - JAIDA/JAS3, Anaphe, OpenScientist, PAIDA (others?)

# Event Generation

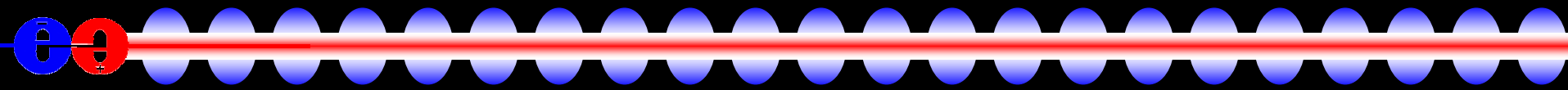




- Still a messy area → not much integration
  - Generation usually based on custom compiled code using a particular FORTRAN or C++ library
  - Different HEPEVT interpretation depending on generators
  - Some generators more “friendly” than others
- Physics generators
  - Pythia, Pandora-Pythia
  - ISAJET
  - WHIZARD
- Single Particles
  - Geant4 GPS
  - Java Diagnostic Event Generator
- Thoughts
  - Would be very useful to have a wrapper similar to ATLAS EvtGen
    - Python probably most promosing for this (or Java)
  - Usually best to rely on existing (debugged/checked) StdHep files

# Reconstruction: Software Development

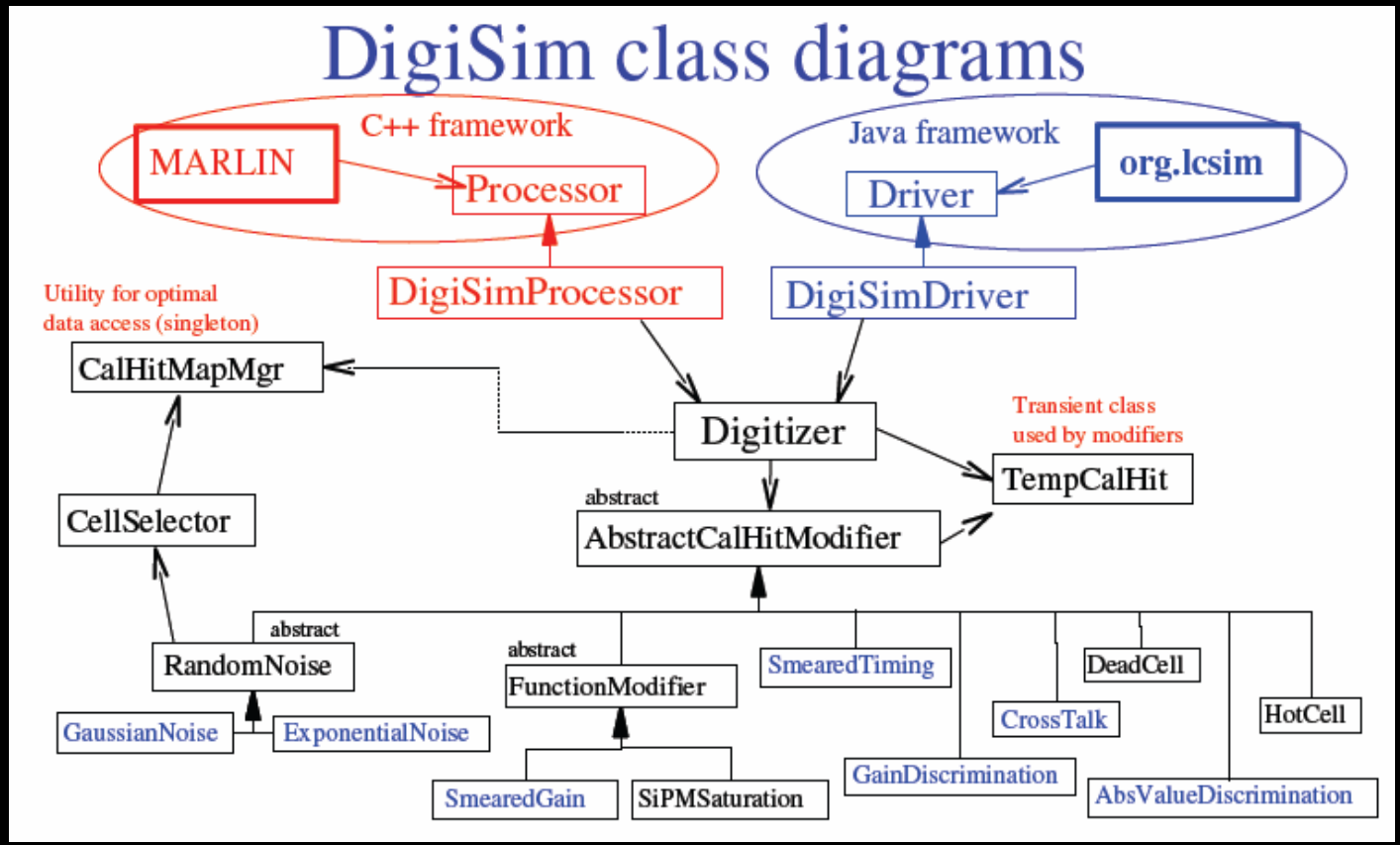




- Required tools
  - Java 1.5
  - Maven
- Recommended tools
  - Netbeans
  - Tortoise CVS (Windows)
- CVS account
  - Contact tony_johnson@slac.stanford.edu
- SLAC CVS projects
  - lcsim, GeomConverter, LCDetectors, SlicDiagnostics, freehep, etc.
- org.lcsim contrib area
  - WIP, personal, or non-compiling codes

# Example Project: DigiSim



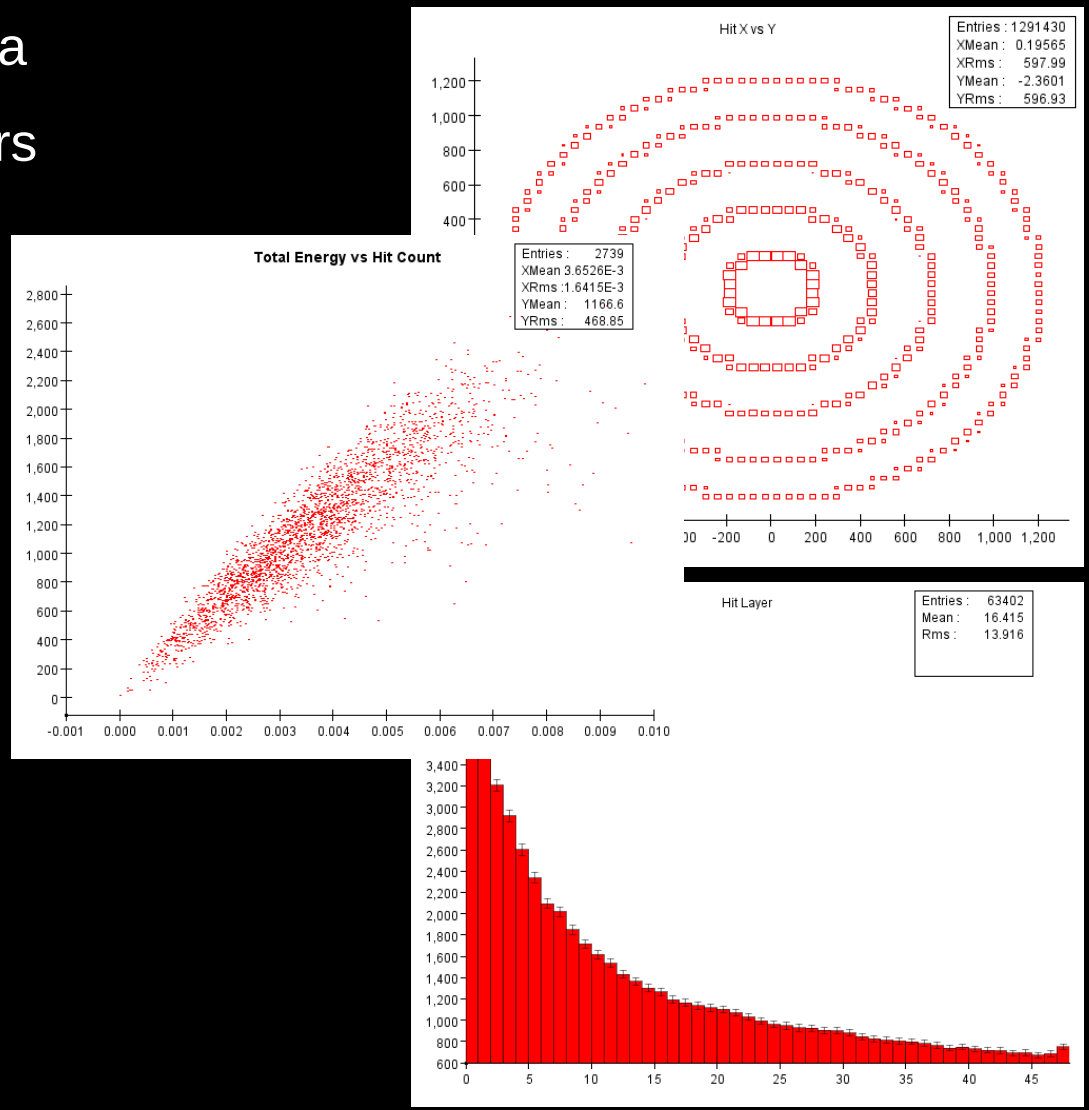



- Goal: a program to parametrically simulate the signal propagation and digitization processes for the ILC detector simulation
- Author: Guilherme Lima

# Example Project: SlicDiagnostics



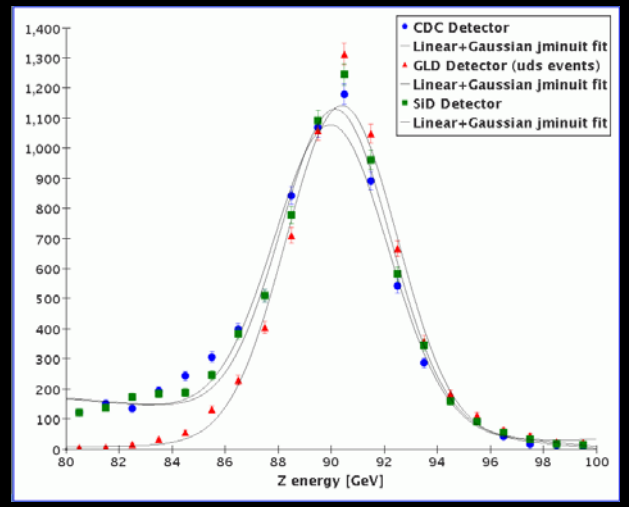
- Diagnostic plots of event data
  - MCParticles, hits, clusters
- Run on different detectors
- Easy to use and setup
  - Maven project
- SLAC CVS project
  - SlicDiagnostics
- Author: Jeremy McCormick

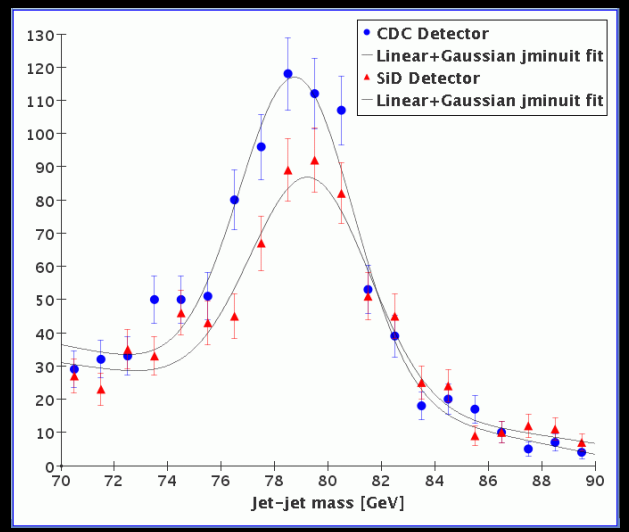


# Example Project: Recon Cheater



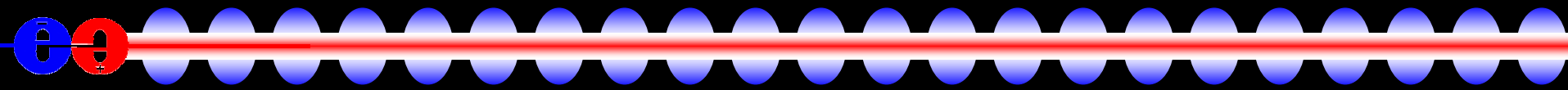
- Makes reconstruction objects from event data
  - Tracks
  - Clusters, refined clusters
  - Reconstructed particles
- Benchmark
  - Detector designs
  - Compare reconstruction algorithms
- Drivers
  - ClusterCheater
  - TrackingCheater
- Customizable using CheatingTable conditions
- Author: Mike Ronan



# Interoperability: Analysis

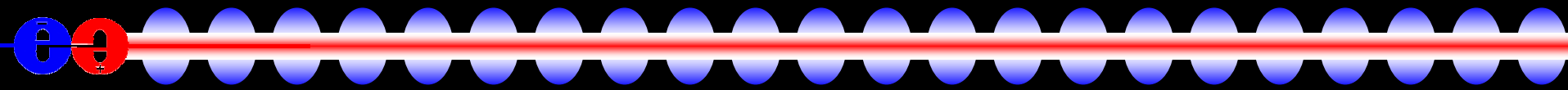




- Same analysis can be run on LCIO files from different simulators
  - Compare performance of detectors
    - e.g. - plot overlays
  - Cross checks
    - Physics
    - Geometries
    - LCIO output
  - Generate LCIO reconstruction objects from different simulator data
- Plot data interchange using .aida files
- JAS3 can also read/write ROOT and PAW files using Freehep libs.
- Some problems with decoding IDs from non-lcsim detectors
  - Working on this!
  - Probably should add more Id'ing metadata to LCIO format

# JAS3

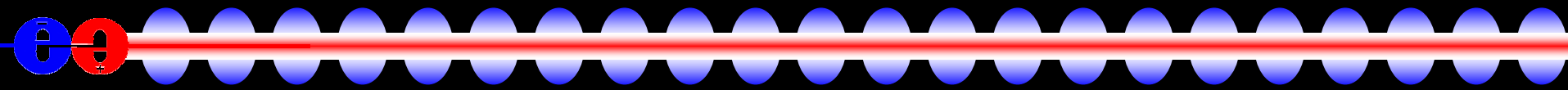




- Physics analysis environment
  - Additional functionality with plugins
- Iterative, event-based analysis model
  - quick development, debugging, ad hoc analysis
- Dynamically load / unload Java analysis drivers
  - From JAR files in the classpath
  - Written and compiled by user
- Plotting engine
  - 1D, 2D histograms, clouds, profiles
  - Output to PNG, JPG, WMF, PS, etc.
- Integrated event display

# Event Display: Screenshot





WIRED: Clusters

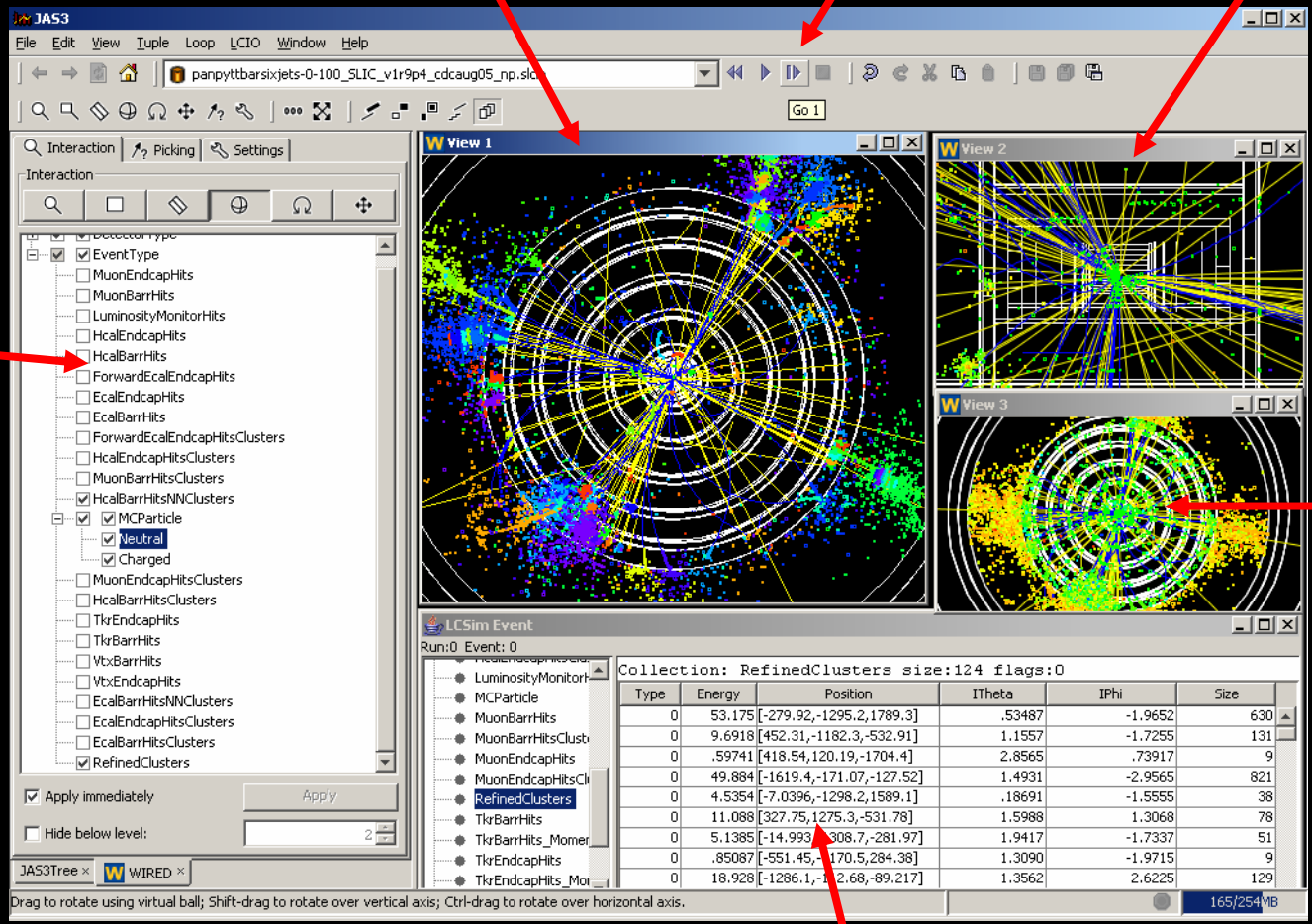
JAS3: Event Loop

WIRED: ZX View

WIRED: LCIO Objects

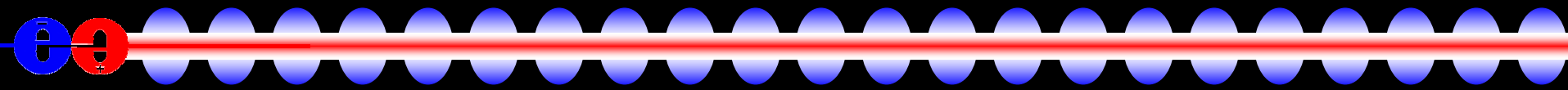
WIRED: Beam View

org.lcsim: LCSim Event Browser



# Reconstruction: Tracking

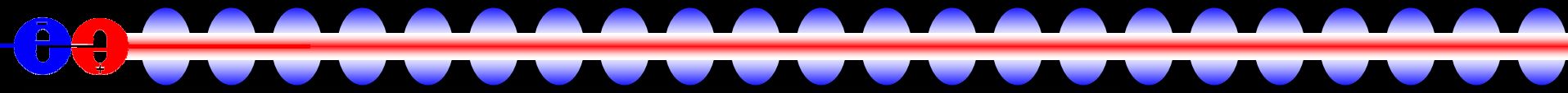


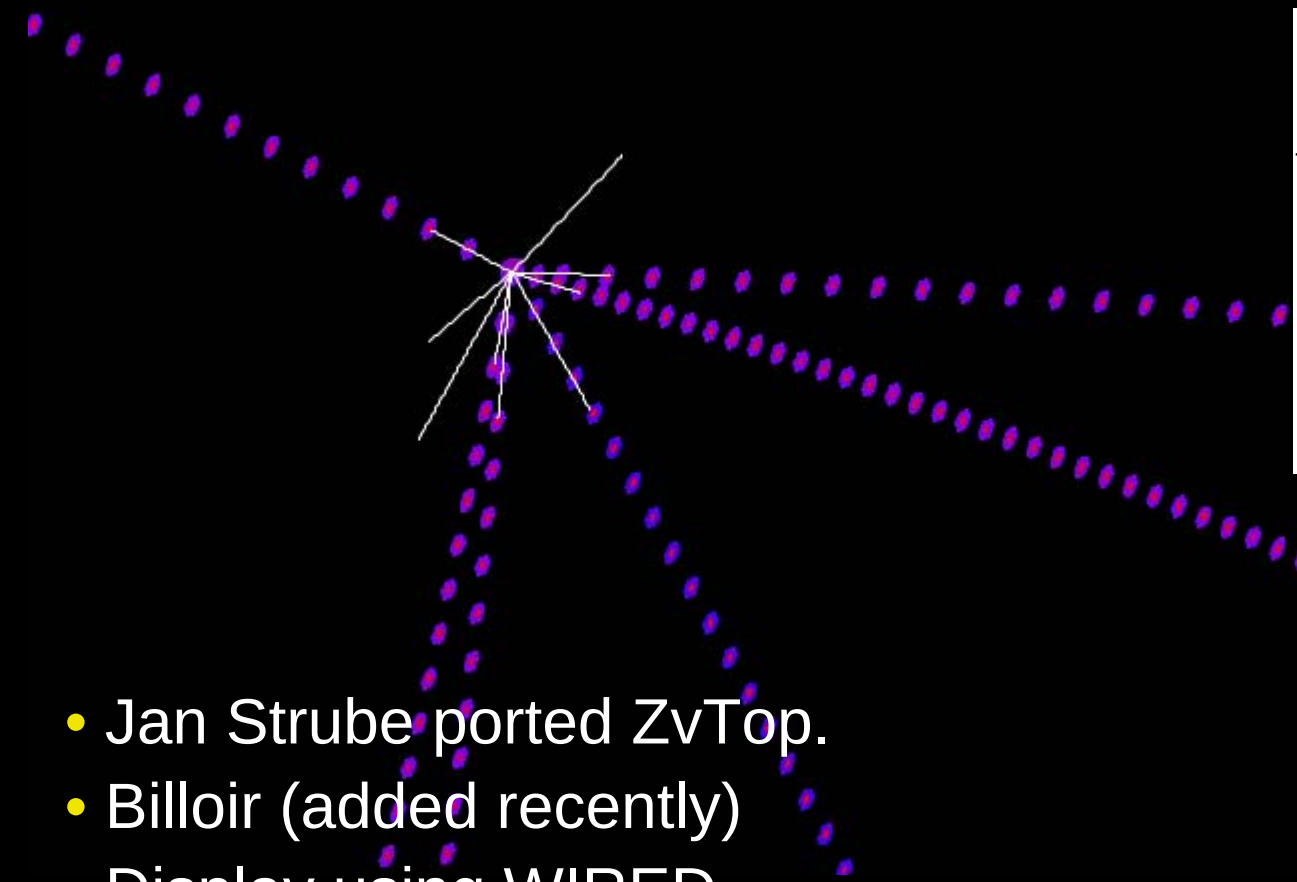


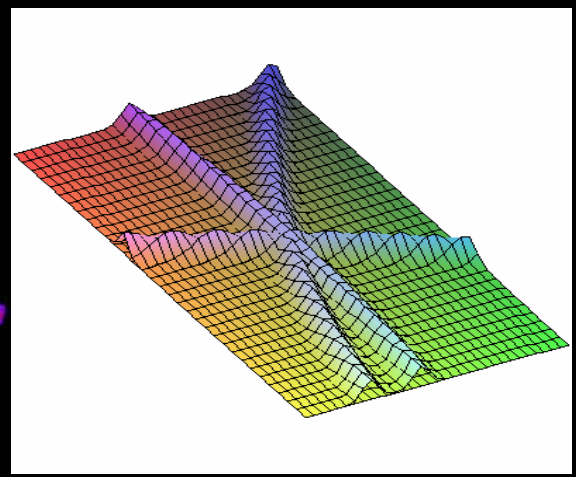
- Most US reconstruction algorithms still using cheater or MC based tracking
- Cheaters
  - MC Fast
  - Mike Ronan's cheater reco package
  - Or just use MCParticles
- Lots of good tools available in org.lcsim for real tracking algo
  - Norman Graf's track fitting and finding
  - Garfield for TPC
  - Nick Sinev's CCD reconstruction
  - Probably a lot of stuff not in org.lcsim CVS, yet (?)
  - Just need to put together into reconstruction algorithms

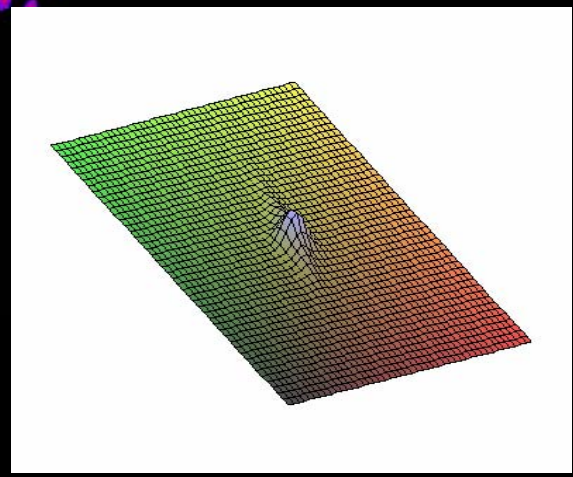
# Reconstruction: Vertexing











- Jan Strube ported ZvTop.
- Billoir (added recently)
- Display using WIRED
- Still not integrated into reconstruction

# Reconstruction: Clustering Algorithms



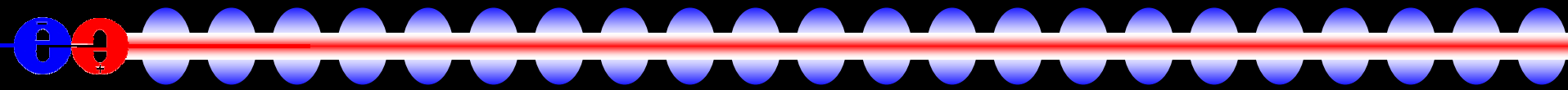
- Clusterers
  - Nearest Neighbor
  - Fixed Cone
  - Directed Tree
  - Min Spanning Tree
  - MIP
  - Cheater
  - contrib/other

```
1  import org.lcsim.event.EventHeader;
2  import org.lcsim.recon.cluster.nn.NearestNeighborClusterDriver;
3  import org.lcsim.recon.cluster.cheat.CheatClusterDriver;
4  import org.lcsim.util.Driver;
5
6  public class ClusterFinding extends Driver
7  {
8          public ClusterFinding()
9          {
10                 int minCells = 5;
11                 add(new NearestNeighborClusterDriver(minCells));
12                 add(new CheatClusterDriver());
13         }
14
15         protected void process(EventHeader event)
16         {
17                 super.process(event);
18         }
19 }
```

- Pick the appropriate clusterer for your analysis
- Run in parallel to compare results
- Utilities and diagnostic plots, also
- Developed by N. Graf, R. Cassell, W. Mader, et al

# Reconstruction: Cluster Display

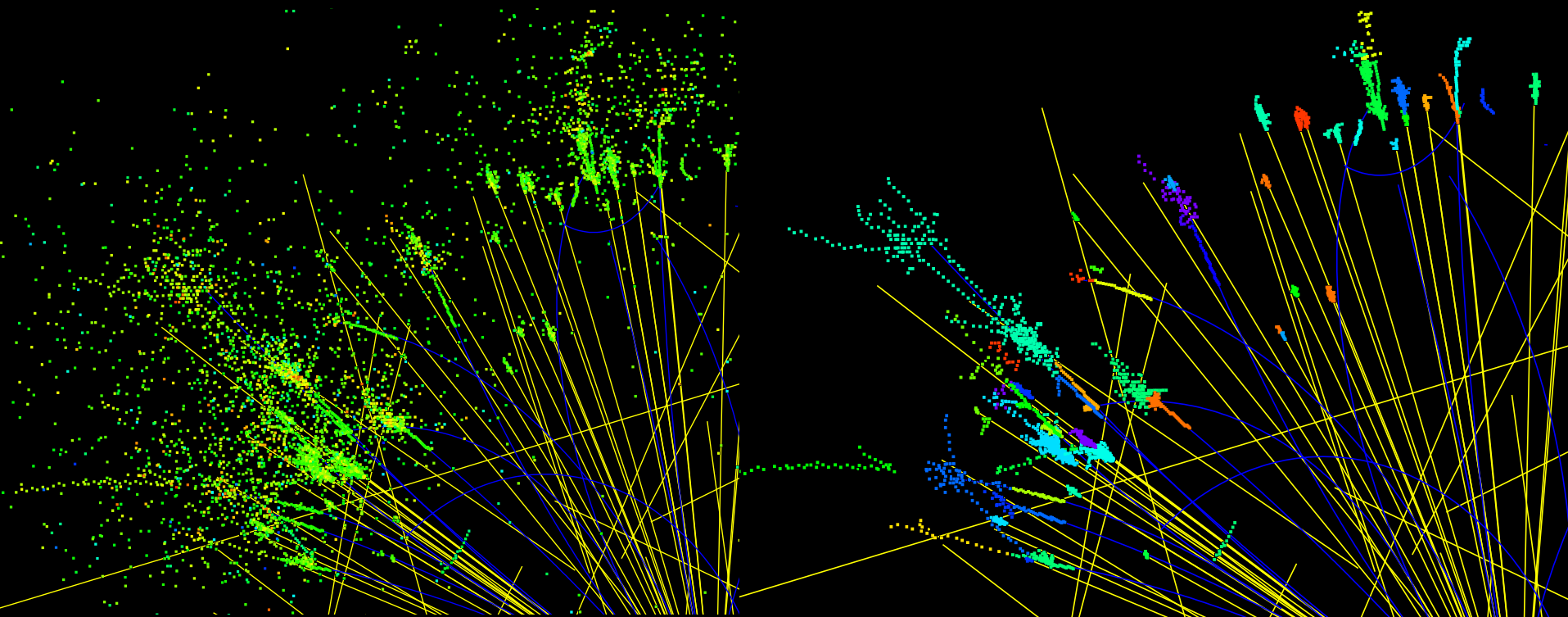




Clusters are displayed automatically by org.lcsim.

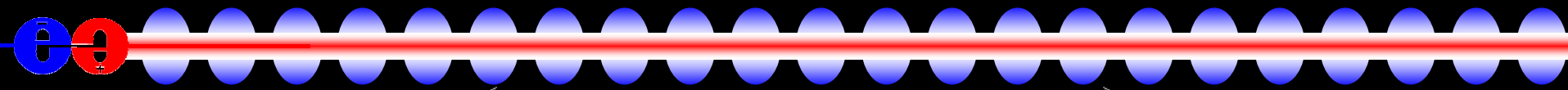
Calorimeter Hits

Nearest Neighbor Clusters
min cells = 5



# Reconstruction: Complete Algorithms





- Lots of people working on full PFA or portions
  - Steve Magill, Norman Graf, Guilherme Lima, Ron Cassell, Matt Charles, Steve Kuhlman, Lei Xia, …
  - See talk by Steve Magill
- Calorimetry part seems to have working PFA algorithms.
- Real tracking algorithms are not included (yet).
- Many utilities and tools exist.
  - Vertex and track fitting, clustering, jet finding
- Need to work on integration