Accessing the Data

Norman Graf (SLAC)

HPS Collaboration Meeting
JLab, D-Day, 2013
LCIO

- Event Data Model and binary persistency format
- Identify the key elements for an event data model appropriate to an HEP experiment.
- Target a simple IO format
  - provide reference implementations in several languages
  - document it well enough to ensure future readability
- Keep the event data model and IO separate
- KISS
  - “Simplify, simplify, simplify” Thoreau
  - “Make everything as simple as possible, but not simpler.” Einstein
LCIO Motivation

LCIO Persistency Framework

Generator

Java, C++, Fortran
Geant3, Geant4

Simulation

Java, C++, Fortran

Reconstruction

Analysis

Java, C++, Fortran

geometry
I have an LCIO File. Now what?

- Icio command-line tool
- Java Analysis Studio (JAS3)
  - LCIO event browser
  - Wired event display
- org.lcsim
  - full access to the event data and geometry
  - Drivers give full access to reconstruction and analysis
  - Loadable within JAS3
  - output LCIO file or AIDA histograms/tuples
- root access via LCIO dictionary
- root access via pyroot
- python access via pyLCIO
lcio Command-Line Tool

> lcio
usage: LcioCommandLineTool
Commands:
count
siodump
select
merge
compare
random
stdhep
run
concat
validate
print
size
split
-h  Print lcio command-line tool usage.
-v  Set the verbosity.
JAS3 LCIO Event Browser

Columns are sortable
JAS3 LCIO Event Browser

Collection: SVTRawTrackerHits size: 101 flags: 80000000
ReadoutName: TrackerHits

Collection: SVTShapeFitParameters size: 101 flags: 80000000

Table:

<table>
<thead>
<tr>
<th>index</th>
<th>nInt</th>
<th>intValues</th>
<th>nFloat</th>
<th>floatValues</th>
<th>nDouble</th>
<th>doubleValues</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-14.010, 1.2576, 1593.2, 32.679, 49.279, NaN, 140.59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-14.822, 1.1737, 1641.9, 29.537, 49.989, NaN, 197.71</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-17.088, 1.2535, 1636.6, 28.580, 50.094, NaN, 187.41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-14.012, 1.0621, 1734.9, 29.069, 50.122, NaN, 179.34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-14.390, 1.2555, 1552.9, 29.644, 50.491, NaN, 202.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-17.494, 1.2527, 1668.9, 28.542, 50.178, NaN, 149.71</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-15.052, 1.1393, 1667.7, 28.841, 49.827, NaN, 204.42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-14.458, 1.2065, 1553.5, 28.983, 50.503, NaN, 144.41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-14.827, 1.1447, 1670.9, 29.306, 49.846, NaN, 206.43</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-13.357, 1.2458, 1510.7, 30.751, 49.773, NaN, 236.87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-16.331, 1.2500, 1612.7, 29.089, 50.674, NaN, 199.41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-13.067, 1.0593, 1645.9, 28.648, 50.319, NaN, 221.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-14.435, 1.1500, 1649.0, 29.430, 50.596, NaN, 165.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-15.660, 1.2457, 1517.8, 28.821, 50.367, NaN, 184.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-14.682, 1.2271, 1554.0, 29.201, 50.426, NaN, 187.42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-13.472, 1.1135, 1622.2, 29.166, 50.324, NaN, 229.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-15.372, 1.0765, 1766.5, 28.801, 50.260, NaN, 244.68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-13.869, 1.2304, 1527.2, 29.644, 50.827, NaN, 154.41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-13.552, 1.1262, 1604.8, 29.056, 50.389, NaN, 152.14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
JAS3 LCIO Event Browser

Collection: SVTRawTrackerHits size:101 flags:80000000

<table>
<thead>
<tr>
<th>index</th>
<th>nInt</th>
<th>nrtValues</th>
<th>nFloat</th>
<th>nFloatValues</th>
<th>nDouble</th>
<th>doubleValues</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-14.010,1.2576,1593.2,32.679,49.279,NaN,140.59</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-14.822,1.1737,1641.9,29.537,49.969,NaN,197.71</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-17.088,1.2353,1639.6,28.590,50.094,NaN,187.41</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-14.117,1.6271,1734.9,29.009,50.122,NaN,179.34</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-14.390,1.2556,1522.9,29.644,50.342,NaN,201.10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-17.494,1.2527,1668.5,28.542,50.178,NaN,149.71</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-15.052,1.1393,1667.7,28.841,49.827,NaN,204.42</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-14.458,1.2065,1553.5,28.963,50.403,NaN,144.41</td>
<td></td>
</tr>
</tbody>
</table>

Size: 0

Collection: SVTShapeFitParameters size:101 flags:80000000

<table>
<thead>
<tr>
<th>index</th>
<th>nInt</th>
<th>nrtValues</th>
<th>nFloat</th>
<th>nFloatValues</th>
<th>nDouble</th>
<th>doubleValues</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-14.010,1.2576,1593.2,32.679,49.279,NaN,140.59</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-14.822,1.1737,1641.9,29.537,49.969,NaN,197.71</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-17.088,1.2353,1639.6,28.590,50.094,NaN,187.41</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-14.117,1.6271,1734.9,29.009,50.122,NaN,179.34</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-14.390,1.2556,1522.9,29.644,50.342,NaN,201.10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-17.494,1.2527,1668.5,28.542,50.178,NaN,149.71</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-15.052,1.1393,1667.7,28.841,49.827,NaN,204.42</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>-14.458,1.2065,1553.5,28.963,50.403,NaN,144.41</td>
<td></td>
</tr>
</tbody>
</table>

Size: 0

Collection: HelicalTrackHits size:6 flags:0

<table>
<thead>
<tr>
<th>Position</th>
<th>CovMatrix</th>
<th>dEdx</th>
<th>Time</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23.360, -34.069, 305.97]</td>
<td>400.54, 19.590, 1.9657, -12.473, -63314, 38895</td>
<td>1.3834E-5</td>
<td>22.056</td>
<td>3</td>
</tr>
</tbody>
</table>
JAS3 LCIO Event Browser

Collection: MCParticle size:12 flags:0

<table>
<thead>
<tr>
<th>N</th>
<th>PDG ID</th>
<th>Type</th>
<th>Generator Status</th>
<th>Simulator Status</th>
<th>Parent</th>
<th>Energy</th>
<th>Momentum</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>22</td>
<td>gamma</td>
<td>Final State</td>
<td>Left</td>
<td></td>
<td>2.0152E-4</td>
<td>5.8564E-5, 5.7878E-6, 1.9243E-4</td>
<td>0.0000, 0.0000, 0.0000</td>
<td>773.86, 75.195, 250.8</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>gamma</td>
<td>Final State</td>
<td>Left</td>
<td></td>
<td>2.7059E-4</td>
<td>-1.7903E-5, 5.2158E-6, 4.8000E-5</td>
<td>0.0000, 0.0000, 0.0000</td>
<td>-173.38, 2500.0, 973.6</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>gamma</td>
<td>Final State</td>
<td>Left</td>
<td></td>
<td>2.5019E-4</td>
<td>-4.6170E-5, 5.0503E-5, 5.2450E-5</td>
<td>0.0000, 0.0000, 0.0000</td>
<td>-47.112, 515.34, 250.0</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>gamma</td>
<td>Final State</td>
<td>Left</td>
<td></td>
<td>2.5334E-4</td>
<td>2.0027E-9, -9.1344E-5, 5.1254E-5</td>
<td>0.0000, 0.0000, 0.0000</td>
<td>2500.0, -1140.3, 156.0</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>gamma</td>
<td>Final State</td>
<td>Left</td>
<td></td>
<td>2.5333E-4</td>
<td>1.6389E-4, 4.1322E-5, 5.1972E-4</td>
<td>0.0000, 0.0000, 0.0000</td>
<td>2125.9, 171.05, 250.0</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
<td>gamma</td>
<td>Final State</td>
<td>Left</td>
<td></td>
<td>2.5019E-4</td>
<td>-1.2756E-4, 4.7036E-5, -5.0340E-5</td>
<td>0.0000, 0.0000, 0.0000</td>
<td>-1567.8, 864.8, 250.0</td>
</tr>
<tr>
<td>6</td>
<td>22</td>
<td>gamma</td>
<td>Final State</td>
<td>Left</td>
<td></td>
<td>2.5028E-4</td>
<td>1.3942E-4, -2.0023E-5, 4.5575E-5</td>
<td>0.0000, 0.0000, 0.0000</td>
<td>1740.8, -2500.0, 696.0</td>
</tr>
<tr>
<td>7</td>
<td>22</td>
<td>gamma</td>
<td>Final State</td>
<td>Left</td>
<td></td>
<td>2.7065E-4</td>
<td>6.6857E-9, -1.6682E-5, 4.1158E-4</td>
<td>0.0000, 0.0000, 0.0000</td>
<td>81.031, 1992.5, 250.0</td>
</tr>
<tr>
<td>8</td>
<td>22</td>
<td>gamma</td>
<td>Other (0)</td>
<td></td>
<td></td>
<td>1.0000</td>
<td>0.0000, 0.0000, 0.0000, 0.0000</td>
<td>0.0000, 0.0000, 1.0000</td>
<td>0.0000, 0.0000, 1.0000</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>e-</td>
<td>Final State</td>
<td>Left</td>
<td></td>
<td>0.27197</td>
<td>0.0036406, -0.0044010, 2.72719</td>
<td>0.0000, 0.0000, 1.0000</td>
<td>-2500.0, -75.298, 249.7</td>
</tr>
<tr>
<td>10</td>
<td>-11</td>
<td>e+</td>
<td>Final State</td>
<td>Decayed In Calorimeter</td>
<td>3.7240</td>
<td>1.19973, 13236, 3.7163</td>
<td>0.0000, 0.0000, 1.0000</td>
<td>182.76, 55.937, 143.9</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>e-</td>
<td>Final State</td>
<td>Stopped</td>
<td></td>
<td>2.6040</td>
<td>0.0024298, -1.1742, 2.6014</td>
<td>0.0000, 0.0000, 1.0000</td>
<td>-155.83, -50.016, 144.1</td>
</tr>
</tbody>
</table>
Wired Event Display

Graphical elements are pickable.
Attributes can be queried.
JAS3 + org.lcsim + Wired + AIDA

- Can also load and execute analysis Drivers from within JAS3
  - Collections added to the Event can be viewed in Wired or inspected in the LCIO event browser
- Support for scripting with full access to org.lcsim
  - jython
  - pnuts
- AIDA histograms and tuples can be created, analyzed, written out for further analysis, etc.
  - AIDA histograms can also be written out in native root format.
Accessing LCIO from root

- ROOT dictionaries available for LCIO since v2
- Compile LCIO with BUILD ROOTDICT=ON
- Start interactive root session, load the dictionaries and use LCIO classes

```cpp
gSystem->load("$LCIO/lib/liblcio.so");
gSystem->load("$LCIO/lib/liblcioDict.so");
IO::LCReader* reader = IOIMPL::LCFactory().getInstance().createLCReader();
reader->open("test.slcio");
EVENT::LCEvent* event = reader->readNextEvent();
while (event) {
    std::cout << event->getRunNumber() << std::endl;
    event = reader->readNextEvent();
}
reader->close();
```
Accessing LCIO from pyroot

- If ROOT is installed with python we get the LCIO python bindings for free!
- Compile LCIO with BUILD ROOTDICT=ON
- Start interactive python session, import ROOT, load the dictionaries and use LCIO classes

```python
from ROOT import gSystem
gSystem.load("$LCIO/lib/liblcio.so")
gSystem.load("$LCIO/lib/liblcioDict.so")
from ROOT import IOIMPL
reader = IOIMPL.LCFactory().getInstance().createLCReader()
reader.open("test.slcio")
event = reader.readNextEvent()
while event:
    print event.getEventNumber()
    event = reader.readNextEvent()
reader.close()
```
Automatic Loading of Dictionaries

- pyLCIO package adds automatic loading of dictionaries on import
- Transparent import of the LCIO namespaces (identical to import from ROOT)
Automatic Loading of Dictionaries

- pyLCIO package adds automatic loading of dictionaries on import
- Transparent import of the LCIO namespaces (identical to import from ROOT)

```python
from pyLCIO import IOIMPL
reader = IOIMPL.LCFactory().getInstance().createLCReader()
reader.open("test.slcio")
event = reader.readNextEvent()
while event:
    print event.getEventNumber()
    event = reader.readNextEvent()
reader.close()
```
Making LCIO Objects Iterable

- Add proper \_\_iter\_\() method to relevant LCIO classes on import
Add proper `__iter__` method to relevant LCIO classes on import

```python
from pyLCIO import IOIMPL
reader = IOIMPL.LCFactory().getInstance().createLCReader()
reader.open("test.slcio")
for event in reader:
    print event.getEventNumber()
reader.close()
```
Making LCIO Objects Iterable

- Add proper `__iter__()` method to relevant LCIO classes on import

```python
from pyLCIO import IOIMPL
reader = IOIMPL.LCFactory().getInstance().createLCReader()
reader.open("test.slcio")
for event in reader:
    print event.getEventNumber()
reader.close()
```

- LCEvent acts like a list of tuples

```python
for collectionName, collection in event:
    print collectionName, collection.getNumberElements()
```
Making LCIO Objects Iterable

- Add proper `__iter__()` method to relevant LCIO classes on import

```python
from pyLCIO import IOIMPL
reader = IOIMPL.LCFactory().getInstance().createLCReader()
reader.open("test.slcio")
for event in reader:
    print event.getEventNumber()
reader.close()
```

- LCEvent acts like a list of tuples

```python
for collectionName, collection in event:
    print collectionName, collection.getNumberofElements()
```

- LC Collection acts like a list

```python
for element in collection:
    print element
```
Enhanced Object Interfaces

- Avoid use of c-style arrays, e.g. `double []`
- Decorate LCIO classes automatically on import depending on existing methods
- Add `getVariableVec()` that returns `TVector3` for all methods that return `double [3]`, e.g. `getPositionVec()` for all classes with `getPosition()`
- Similarly add `setVariableVec(TVector3)` to all IMPL classes
- Add `getLorentzVec()` that returns `TLorentzVector` to all classes that support `getMomentum()` and `getEnergy()`
IO::LCReader and UTIL::LCStdHepRdr now fulfill iterable interface
Reading LCIO and StdHep Files

- IO::LCReader and UTIL::LCStdHepRdr now fulfill iterable interface
- Offer two wrapper classes that streamline the interface of both readers
- Allow transparent loop over all input files

```python
from pyLCIO.io.StdHepReader import StdHepReader
reader = StdHepReader("test.stdhep")
reader.addFile("test2.stdhep")
reader.addFiles(['test3.stdhep', 'test4.stdhep'])
reader.addFileList("stdhepFiles.txt")
reader.skip(10)
for event in reader:
    print event.getEventNumber()
reader.close()
```
Event Loop and Analysis

- Provide a managed event loop similar to Marlin/org.lcsim
- Plug in user classes that are executed for each event
- Support LCIO and StdHep input using the new reader interface
- File type handled by event loop: `eventLoop.setFile( fileName )` independent of file type (determined by file extension)
- User class must inherit from Driver or implement: `startOfData()`, `process( event )`, `endOfData()`
Example Driver

```python
from pyLCIO.drivers.Driver import Driver
from ROOT import TH1D, TCanvas

class McParticlePlotDriver(Driver):
    def __init__(self):
        Driver.__init__(self)
        self.histograms = {}

    def startOfData(self):
        self.histograms['Energy'] = TH1D('Energy', 'Energy; Energy [GeV]; Entries', 50, 0., 260.)
        self.histograms['Pt'] = TH1D('Pt', 'pT;p_T [GeV];Entries', 50, 0., 100.)
        self.histograms['PDGID'] = TH1D('PDGID', 'PDG ID;PDG ID;Entries', 1200, -600, 600.)
        self.histograms['GeneratorStatus'] = TH1D('GeneratorStatus', 'Generator Status;Generator Status;Entries')

    def processEvent(self, event):
        mcParticles = event.getMcParticles()
        for mcParticle in mcParticles:
            v = mcParticle.getLorentzVec()
            self.histograms['Energy'].Fill(v.Energy())
            self.histograms['Pt'].Fill(v.Pt())
            self.histograms['PDGID'].Fill(mcParticle.getPDG())
            self.histograms['GeneratorStatus'].Fill(mcParticle.getGeneratorStatus())

    def endOfData(self):
        plots = []
        for histogramName in self.histograms:
            plot = TCanvas('c%s' % histogramName, histogramName)
            self.histograms[histogramName].Draw()
            plots.append(plot)

        userInput = raw_input('Press any key to continue')
```

`$LCIO/examples/python/exampleDrivers/McParticlePlotDriver.py`
from pyLCIO.base.EventLoop import EventLoop
from pyLCIO.drivers.EventMarkerDriver import EventMarkerDriver
from exampleDrivers.McParticlePlotDriver import McParticlePlotDriver
import sys, os

def McParticlePlots(fileName):
    eventLoop = EventLoop()
    # Set the input file. The actual reader is determined from the file ending (stdhep or slcio)
    eventLoop.setFile(fileName)
    # Add a driver to print the progress
    markerDriver = EventMarkerDriver()
    markerDriver.setInterval(1)
    markerDriver.setShowRunNumber(False)
    eventLoop.add(markerDriver)
    # Add the driver that draws the MCParticle plots
    mcParticlePlotDriver = McParticlePlotDriver()
    eventLoop.add(mcParticlePlotDriver)
    # Skip some events if desired
    eventLoop.skipEvents(0)
    # Execute the event loop
    eventLoop.loop(-1)

def usage():
    print 'Usage:
    python %s <fileName>' % ( os.path.split(sys.argv[0])[1] )

if __name__ == '__main__':
    if len(sys.argv) < 2:
        usage()
        sys.exit(0)
    # Read the file name from the command line input
    fileName = sys.argv[1]
    McParticlePlots(fileName)
XML Steering of Drivers (Experimental)

- Provide executable that parses an XML file and sets up the event loop
- At the moment very limited features - can be expanded if there is demand
- Run it: python $LCIO/src/python/pylcio steering.xml

```
<pylcio>
  <inputFiles>
    <file> test.slcio </file>
  </inputFiles>

  <control>
    <skipEvents>0</skipEvents>
    <numberOfEvents>-1</numberOfEvents>
    <printDrivers>True</printDrivers>
    <printStatistics>true</printStatistics>
  </control>

  <execute>
    <driver name="markerDriver"/>
    <driver name="mcParticlePlotDriver"/>
  </execute>

  <drivers>
    <driver name="markerDriver" type="pyLCIO.drivers.EventMarkerDriver.EventMarkerDriver">
      <interval> 1 </interval>
      <showRunNumber> False </showRunNumber>
    </driver>
    <driver name="mcParticlePlotDriver" type="exampleDrivers.McParticlePlotDriver.McParticlePlotDriver"/>
  </drivers>
</pylcio>
```

$LCIO/examples/python/exampleSteering/McParticlePlots.xml
Summary

- Python bindings work out of the box with ROOT LCIO dictionaries through pyROOT
- pyLCIO package for additional features
  - Automatic loading of ROOT and LCIO dictionaries on import
  - Added iterator methods to container and reader classes to allow *pythonic* loops
  - Additional accessor methods for LCIO classes to directly get `TVector3` and `TLorentzVector` where appropriate
  - Wrapper classes for `LCReader` and `LCStdHepRdr` to streamline interface
  - Managed event loop with driver/processor style plug-in of user code
  - XML steering of drivers (experimental)
- Use this for high level tasks like analysis, plotting and creation of tuples/trees
- Can not replace complex reconstruction algorithms in org.lcsim
- (Currently) no geometry information except raw cell IDs stored with hits
Requirements

- Install ROOT with python bindings
- Get LCIO version v02-04 or newer

- Compile with BUILD_ROOTDICT=ON
- Add ROOT and pyLCIO to python environment
  export PYTHONPATH=$LCIO/src/python:$PYTHONPATH
- Or simply source $LCIO/setup.sh
The data are out there…

- LCIO provides the Event Data Model and persistence format for HPS data.
  - Well documented API and binary IO format.

- Multiple language bindings and multiple toolkits exist to provide access to the data.

- Encourage all of you to look at the data and provide feedback on both the data quality and the functionality of the data analysis tools.

- Get involved!
Further Information

- Confluence wiki page at: https://confluence.slac.stanford.edu/display/hpsg/Simulation+and+Reconstruction+Software
- HPS Software mailing list: hps-software@slac.stanford.edu
- Issues/Bug tracker: https://jira.slac.stanford.edu/browse/HPSJAVA/