

Can I use org.lcsim to filter stdhep files?

When reading a stdhep file with the org.lcsim classes the events are immediately converted into org.lcsim "EventHeader" objects, however a pointer to the original stdhep file is kept, so it is possible to write the original event back out as a stdhep file.

The example below shows a Driver which will write out only events with >100 MC particles:

StdhepFilter.java

```
import org.lcsim.event.EventHeader;
import org.lcsim.util.Driver;
import org.lcsim.util.loop.StdhepDriver;

public class StdhepFilter extends Driver
{
    public StdhepFilter()
    {
        // Add a child driver which will write out the event
        add(new StdhepDriver("myfile.stdhep", "My Title", "My Comment", 0));
    }
    protected void process(EventHeader event)
    {
        int n = event.getMCParticles().size();
        if (n>100)
        {
            // If the event is accepted call super.process() which calls the
            // child drivers, and thus writes out the event
            super.process(event);
        }
    }
}
```

Alternative approach

If you only want to make trivial cuts on a stdhep file, then it may be more efficient to use the stdhep reading/writing classes directly. There are two representations of stdhep events available, a true Object-Oriented approach using Particle and Event objects, and a lower level interface which reflects the Stdhep "common-block" format. If you want to read stdhep files, apply a simple filter, and write them out again, the "common-block" style interface, while a little less pretty, is probably considerably more efficient.

The following program can be compiled and run in JAS3 if you have the org.lcsim plugin installed. You will need to modify the input and output file specifications.

StdhepFilter.java

```
import hep.io.stdhep.StdhepEvent;
import hep.io.stdhep.StdhepReader;
import hep.io.stdhep.StdhepRecord;
import hep.io.stdhep.StdhepWriter;
import java.io.EOFException;
import java.io.IOException;

/**
 * A simple class for filtering stdhep events
 */
public class StdhepFilter
{
    public static void main( String[] args ) throws IOException
    {
        int nRecord = 0;
        int nIn = 0;
        int nOut = 0;

        StdhepReader reader = new StdhepReader("inputfile.stdhep");
        StdhepWriter writer = new StdhepWriter("outputfile.stdhep",reader.getTitle(),reader.getComment(),0);

        try
        {
            for (;;)
            {
                StdhepRecord record = reader.nextRecord();
                nRecord++;
                if (record instanceof StdhepEvent)
                {
                    StdhepEvent event = (StdhepEvent) record;
                    nIn++;
                    // Insert your filter code here!
                    if (event.getNHEP() < 100) continue;
                }
                writer.writeRecord(record);
                nOut++;
            }
        }
        catch (EOFException x)
        {
            // No problem
        }
        finally
        {
            writer.close();
            reader.close();
            System.out.println("Records: "+nRecord+" Events: "+nIn+" Out: "+nOut);
        }
    }
}
```

Documentation on the freehep-stdhep classes is available here:

<http://java.freehep.org/freehep-stdhep/apidocs/index.html>

Documentation on the freehep-physics classes (only necessary if you want the more OO access to stdhep events) is here:

<http://java.freehep.org/freehep-physics/apidocs/index.html>

Note in particular the [StdhepConverter](#) class.