

Tutorial 1 FastSim and Jet Clustering

This is a short example on how to run the MCFast tool, generate a list of reconstructed particles and run a Jet Cluster

Requirements

1. JAS3 must be installed
2. Get a stdhep file, [Z Pole stdhep](#) from the SLAC HEP repository
3. Select sid01 as detector version when JAS3 prompts you

The Code

```

import java.io.File;
import java.util.List;
import java.util.ArrayList;

import org.lcsim.mc.fast.MCFast;
import org.lcsim.util.Driver;
import org.lcsim.util.loop.LCIODriver;
import org.lcsim.event.EventHeader;
import org.lcsim.event.ReconstructedParticle;
import org.lcsim.event.util.JetDriver;

import hep.physics.jet.*;

public class SimpleEventSim extends Driver
{
    int ievt=0;
    String LCIOOutputfilename="fastmcZuds.slcio";
    String jetlistname = "Jets";

    public SimpleEventSim()
    {
        // Create MCFast with standard options
        Driver fast = new MCFast();
        // Turn on diagnostic histograms
        fast.setHistogramLevel(HLEVEL_NORMAL);
        // Add as sub-driver
        add(fast);
    // set up Jet Finder
        JetDriver j = new JetDriver();
        j.setInputCollectionName("MCFastReconstructedParticles");
        j.setOutputCollectionName(jetlistname);
        JetFinder twojet = new FixNumberOfJetsFinder(2);
        j.setFinder(twojet);
        add(j);

        // write lcio files
        File output = new File(System.getProperty("user.home"),LCIOOutputfilename);
        add(new LCIODriver(output));
    }
    protected void process(EventHeader event)
    {
    // call other driver process methods
        super.process(event);
    // give some status on events
        if (ievt%50==0)
        {
            System.out.println("Processed Events  " + ievt);
        }
        ievt++;
    }
}

```

Loading this code into JAS will run over the stdhep file and generate and output LCIO file for analysis later

MCFAST

```

// Create MCFast with standard options
Driver fast = new MCFast();
// Turn on diagnostic histograms
fast.setHistogramLevel(HLEVEL_NORMAL);
// Add as sub-driver
add(fast);

```

This initializes the fast Monte-Carlo Generator and adds it to the event loop

JET Clustering

```
// set up Jet Finder
JetDriver j = new JetDriver();
j.setInputCollectionName("MCFastReconstructedParticles");
j.setOutputCollectionName(jetlistname);
JetFinder twojet = new FixNumberOfJetsFinder(2);
j.setFinder(twojet);
add(j);
```

The JetDriver needs an input and output collection and a separate JetFinder, which in this example is a FixNumberOfJetsFinder finder. The JetFinder Object is then added to the JetDriver. For details see the [FreeHep JetFinder](#) documentation.

LCIO output

```
// write lcio files
File output = new File(System.getProperty("user.home"),LCIOoutputfilename);
add(new LCIODriver(output));
```

This opens a new file and writes all LCIO Collections into the file.