

# Simulation and Reconstruction

- [Overview](#)
  - [Required Software](#)
- [Event Generation With MadGraph and Pythia](#)
  - [Download and Install MadGraph5 and Pythia](#)
  - [Generate Events:](#)
- [Installing ILCSoft programs with ilcinstall](#)
  - [Installing SLIC](#)
  - [Installing others](#)
- [Running SLIC](#)
- [Event Reconstruction with PandoraPFA](#)

## Overview

This page will guide you through the process of installing and using some of the software we use to generate, fully simulate and reconstruct events in the MCDRCal01 detector for a Higgs Factory Muon Collider. We assume a Linux environment.

We start with the MadGraph5 matrix element generator, a simple program that allows you to define a process or group of processes, calculate their cross sections, generate events and seamlessly interface with Pythia for Monte Carlo shower simulation (hadronization).

We then fully simulate these events in our detector with SLIC and reconstruct them with algorithms and frameworks provided by PandoraPFA, LCFIVertex and org.lcsim.

(Note: This page is currently under construction and is not yet complete)

## Required Software

For this tutorial you will need to have the following programs already installed:

- Java
- Python 2
- SVN
- CVS
- ROOT

**Note:** The installation of one slicPandora dependency (Gear) will require the environment variable \$ROOTSYS to be set, which is not compatible with most distribution-provided ROOT packages. If you have trouble with the installation, install ROOT with the 'Location Independent Installation' method described [here](#). For more info on this issue, see [this thread](#).

Other ilcSoft packages not covered in this tutorial will require installations of

- MYSQL
- CERNLIB

## Event Generation With MadGraph and Pythia

**Note:** If you already have the .hep or .stdhep files you want, or if you just want to use the particle gun in SLIC, you do not need to do the steps in this section; go on to Installing ILCSoft Programs with ilcinstall. If you have fully simulated .slcio events, go on to Analysis.

## Download and Install MadGraph5 and Pythia

Download MadGraph 5 from the [MadGraph 5](#) downloads page after making a free account and unpack the file somewhere.

```
$ tar xzvf MadGraph5_v1_5_11.tar.gz
```

Next, use MadGraph to download and install Pythia:

```
$ cd MadGraph5_v1_5_11
$ ./bin/mg5
mg5> install pythia-pgs
```

**Note:** If you have Python 3 installed on your system you may need to run MadGraph and associated programs with Python 2, eg.

```
$ python2 ./bin/mg5
```

## Generate Events:

For this tutorial we will be simulating the process of two muons colliding with a center of mass energy of 125 GeV, creating a  $Z^*$  boson which decays into two light quarks. This data will be used to generate detector-specific information that is used by the PandoraPFA photon reconstruction algorithm. A copy of this 'likelihood data' will be provided for the mcdrcal01 detector. For a more complete introduction to MadGraph, simply type `mg5> tutorial` or refer to the [MadGraph 5 Wiki](#).

```
mg5> define uds = u d s u~ d~ s~      # define the multiparticle uds as any one of the u, d and s quarks or
their antiparticles
mg5> generate mu+ mu- > Z > uds uds    # calculate the diagrams for the process
                                     # use, eg., `add process mu+ mu- > Z > b b~` to add another processes.
mg5> output my-mumu-z-uds              # create a folder in the current directory.
mg5> open index.html                  # optional; view the generated diagrams and other information
```

**Important note:** If you want to do  $\mu^+ \mu^- \rightarrow h \rightarrow x x$  events to load this interaction vertex, be sure to execute `mg5> import model sm-full` before launching. The  $\mu^+ \mu^- \rightarrow h$  vertex is disabled in the default sm model for performance reasons.

Now we can generate events and simulate their hadronization with Pythia.

```
> launch
> 2      # select to run MadEvent and Pythia (select 1 to only calculate cross sections)
> 2      # edit run card. eg. Change beam type (0 for muons or electrons, ie. no PDF),
        # set energy (62.5GeV each), edit cuts, number of events
        # none of the cuts apply to uds quarks, so in this example we will ignore them.
> 1      # (optional) edit parameter card to change physics parameters such as Higgs mass and width.
> 0      # Launches event generator and then runs Pythia.
> exit
$ cd my-mumu-z-uds/
$ gzip -d Events/run_01/tag_1_pythia_events.hep.gz
```

Now we have a .hep file which can be read into SLIC for full detector simulation. If you want to open the events in JAS3, change the extension to `.stdhep` after extracting the file.

## Installing ILCSoft programs with ilcinstall

Now we will use the [ilcinstall](#) program to install the various ilcSoft packages we will need. These will be:

- slic
  - Geant4-based simulation
- slicPandora
  - PandoraPFA event reconstruction algorithms
- PandoraSettings
  - Useful premade settings files for PandoraPFA
- lcplus
  - LCFIPlus reconstruction does vertexing, jet finding and flavor tagging.

ilcinstall is installed with the command

```
$ svn co https://svnsrv.desy.de/public/ilctools/ilcinstall/trunk/ ilcinstall
```

This should create a folder called ilcinstall in the directory from which you ran the command. The version used for this tutorial was v01-17-04, rev 2914. For more help installing and using ilcinstall, see the [ilcinstall documentation](#).

## Installing SLIC

**Note:** Installing SLIC will involve compiling several programs, including Geant4 which is quite large. This will take a while even on a fast machine.

Once you have successfully checked out the ilcinstall project you should be able to install the rest of the tools. Start by opening the file `ilcinstall/examples/slic/release-versions.py` in an editor and change the variables `ilcsoft_install_prefix`, `Java_version` and `Java_path` to reflect your environment:

- `Java_version` can be found by running `$ java -version`. Here we use `1.7.0_40`.
- `Java_path` is the directory containing `bin/java`, which can be found with `$ which java`
- `ilcsoft_install_prefix` will be the name of the directory in which it installs programs, eg. `/home/$USER/ilcsoft/`.

Now you should be ready to install slic. First, check that everything is set up correctly:

```
cd ilcinstall
./ilcsoft-install examples/slic/release-scratch.cfg -p
```

If this finishes with no errors, run the actual install with

```
./ilcsoft-install examples/slic/release-scratch.cfg -i
```

The script should then download, build and install slic and its dependencies.

## Installing others

The same process should work for installing slicPandora, PandoraSettings and lcplus.

If you run into problems compiling various dependencies, check their README's (eg. /home/\$USER/slicPandora/v00-00/gear/v01-03/README) and ensure you have the proper dependencies installed and environment variables set. If you can't figure it out, ask for help in the Marlin et al section of the [Linear Collider Forum](#).

**Note:** At the time of this writing there was an error in the slicPandora install scripts that needed to be fixed by adding the line

```
from slicpandora import SlicPandora
```

to the top of the file examples/slicPandora/release-versions.py (See [this page for more information](#)).

## Running SLIC

(...Link to mcdrcal01 file...)

SLIC Script:

### Z\_uds\_125GeV\_iso

```
/physics/select FTFP_BERT
/physics/setPDGFile /path/to/slic/v00-00/slic/v03-01-04/data/particle.tbl
/lcio/path /path/to/eventdata
/lcio/filename Z_uds_125GeV_iso
/lcio/fileExists delete
/lcdd/url /path/to/detectors/mcdrcal01/mcdrcal01.lcdd
/run/initialize
/vis/open HepRepFile
/vis/heprep/setFileDir /path/to/detectors/mcdrcal01/
/vis/heprep/setFileName mcdrcal01
/vis/heprep/setCullInvisibles true
/vis/scene/create
/vis/sceneHandler/attach
/vis/viewer/set/culling global
/vis/enable
/vis/viewer/flush
/lcdd/dumpGDML /path/to/detectors/mcdrcal01/mcdrcal01.gdml
/generator/select stdhep
/generator/filename /path/to/mumu-Z-uds-125GeV.hep
/run/beamOn 5000
```

(...Running on mcdrd or OSG?...)

```
<mcdrd.fnal.gov> nohup nice -m ~/path/to/slic/script
```

## Event Reconstruction with PandoraPFA

(...Installation of slicPandora and PandoraSettings...)

#### PandoraSettingsDefault.xml excerpt

```
<!-- Standalone photon clustering -->
<algorithm type = "PhotonReconstruction">
  <algorithm type = "ConeClustering" description = "PhotonClusterFormation">
    <ClusterSeedStrategy>0</ClusterSeedStrategy>
    <ShouldUseTrackSeed>false</ShouldUseTrackSeed>
    <ShouldUseOnlyECalHits>true</ShouldUseOnlyECalHits>
    <ConeApproachMaxSeparation>250.</ConeApproachMaxSeparation>
  </algorithm>
  <ClusterListName>PhotonClusters</ClusterListName>
  <ShouldMakePdfHistograms>true</ShouldMakePdfHistograms>
  <NEnergyBins>9</NEnergyBins>
  <EnergyBinLowerEdges>0 0.5 1 1.5 2.5 5 10 20 50 </EnergyBinLowerEdges>
  <HistogramFile>MyPath/MyLikelihoodData9EBins.xml</HistogramFile>
</algorithm>
```

[...Link to LikelihoodData I generated...](#)

## Using WHIZARD 2 to calculate cross sections

At the time of writing, the Standard Model in MadGraph does not include the vertices  $\mu^+ \mu^- \rightarrow h$  or  $e^+ e^- \rightarrow h$  (See [bug report](#)). It is also not a reliable cross section calculator due to the facts that it can only deal with tree-level processes and it does not compute the effects of interference from diagrams that are not explicitly added by the user.

WHIZARD 2 is a good tool for more precise cross section calculations because it computes every possible channel between the given initial and final states. One necessary restriction of this is that unlike MadGraph, you do not have control over the intermediate channels, such as whether your  $\mu^+ \mu^- \rightarrow b \bar{b}$  process is mediated by a Z or h boson. In addition, stdhep files generated by WHIZARD retain little Monte Carlo information about the event. Refer to the website for instructions on installing and using WHIZARD 2. We provide here a sample SINDARIN input file for a muon collider at 3TeV.

#### mumu-nnbbsb.in

```
# SM_ac allows to change higgs tri-linear self-coupling (fac_gh3)
model = SM_ac
process mumu_nnbbsb = "mu-", "mu+" => numu, numubar, h, h
mH = 125 GeV
wH = 4.07 MeV
fac_gh3 = 1.2
sqrts = 3000 GeV
beams = "mu-", "mu+"
n_events = 10000
iterations = 10:10000, 5:10000, 5:50000

simulate (mumu_hhnn) { $sample = "/path/of/output/data/folder" sample_format = stdhep }
```

This SINDARIN script calculates the cross section of double-Higgs production via W-fusion at a high energy muon collider. The cross section is dependent on the Higgs trilinear self-coupling constant, which can be expressed as a ratio to the Standard Model value. In this example the constant is 1.2 times the Standard Model value. Advanced SINDARIN scripts can perform simple analysis and plot histograms. See the [WHIZARD 2 manual](#) for a much more detailed look at the capabilities of WHIZARD.