Table of Contents

- **Links**:
    - ParticleGunForAtlas
    - ParticleGun samplers (python class)

from https://svnweb.cern.ch/trac/atlasoff/browser/Generators/ParticleGun/trunk/python/samplers.py#L852

```
def mksampler(x):
        """
        Automatically cast the provided object to a sampler type. This is used
        extensively inside the particle and position samplers, so that the user
        can pass in a primitive type like a number or list and it will be
        treated as if the more verbose sampler constructors had been called.

        Behaviour:
         - if x can be called, i.e. x() is valid, we just return x;
         - a Python list (square brackets) will be converted to a continuous UniformSampler or Dis
         - a Python tuple (round brackets/parentheses) will be treated as a discrete CyclicSeqSamp
         - a Python set (curly brackets/braces) will be treated as a discrete RandomSeqSampler;
         - otherwise a ConstSampler will be created from x, so that x is returned when the sampler
```

- [ ] = list => continuous UniformSampler (flat sampling in a range) or DisjointUniformSampler (flat sampling in a set of disjoint ranges, e.g. eta crack regions)
- ( ) = tuple => discrete CyclicSeqSampler (choose in increasing order from a discrete sequence of values, with cyclic repetition on the sequence length)
- { } = set => discrete RandomSeqSampler (choose randomly from a discrete sequence of values)

Example: jo to generate a "multimuon" sample, with 10 muons per event, where

- 5 mu+ and 5 mu-
- flat pt distribution [1 GeV; 50 GeV]
- flat eta distribution [-3.0, 3.0]
- flat phi distribution [0; 2pi]

As we are interested in specifying the particle kinermatics in terms of pt, eta and phi (and mass), note the use of the specific PtEtaMPhiSampler for the MonSampler.

```
# jobOptions (to be used with Generate_tf) for multimuon event generation.

#--------------------------------------------------------------
# Configuration for EvgenJobTransforms
#--------------------------------------------------------------
evgenConfig.description = "Multimuon events in MC15"
evgenConfig.keywords = ["muon", "singleParticle"]
evgenConfig.generators = ["ParticleGun"]
evgenConfig.contact = ["Sergio.Gonzalez.Sevilla@cern.ch"]

#--------------------------------------------------------------
# Configuration for ParticleGun
#--------------------------------------------------------------
import ParticleGun as PG
pg = PG.ParticleGun()
pg.randomSeed = 123456

pg.sampler.pid = (13,-13) # cycle mu- (+13) and mu+ (-13)
pg.sampler.n=10
pg.sampler.mom = PG.PtEtaMPhiSampler(pt=[1000,50000], eta=[-3.0,3.])
genSeq += pg
```

LHAPDF (OSX 10.6.7)

- download from http://projects.hepforge.org/lhapdf/
- gfortran is needed: http://gcc.gnu.org/wiki/GFortranBinaries
- run configure: `./configure --enable-low-memory --disable-pyext --prefix=a_path CC=/sw/bin/gcc-4 CXX=/sw/bin/g++-4 FC=/sw/bin/gfortran`
    - even if with different versions of gcc/g++ with respect to gfortran the compilation succeeded (`CC=/usr/bin/gcc-4.2 CXX=/usr/bin/g++-4.2 FC=/usr/local/bin/gfortran=`) , I decided to use coherent binaries overall because the compilation of Whizard with `/usr/local/bin/gfortran` (4.5.1 20100506 (prerelease)) failed but not with `/sw/bin/gfortran` (4.5.0)... even using same compiler flags... no idea why...
    - where for `a_path` I've chosen `/Users/sevilla/software/lhapdf/lhapdf-5.8.5`
    - Because of the increasing numbers and complexity of new PDFs, from version 5.7.0 onwards the memory requirement declared by LHAPDF has exceeded 1GB!. A solution is to build LHAPDF in the "low memory" mode by specifying the `--enable-low-memory` option at configuration.
    - the `--disable-pyext` allows to avoid some compilation errors (`unrecognized command line option "-Wno-long-double"`)
- do `make` followed by `make install`.
- optionally do `make check`, but if no PDF data files have been installed yet, it will fail the test. Also note it will look for data sets within `share/lhapdf/PDFsets`. The environment variable `LHAPATH` can be used to specify an alternative location.

How to install PDF sets

Due to the increasing number and size of PDF data files, LHAPDF no longer bundles PDF set data in the package tarball. The sets are instead all stored online at `http://svn.hepforge.org/lhapdf/pdfsets/5/` and you should install those that you wish to use into `/Users/sevilla/software/lhapdf/lhapdf-5.8.5/share/lhapdf`. The lhapdf-getdata script helps to automate the set download and installation process: before installation you will find it in the `bin` directory. Alternatively, download the sets by hand from the URL given above, and place them in the PDF sets directory.

```
export PATH=/Users/sevilla/software/lhapdf/lhapdf-5.8.5/bin/:${PATH}
cd /Users/sevilla/software/lhapdf/lhapdf-5.8.5/share/lhapdf/PDFsets
lhapdf-getdata cteq5l.LHgrid
lhapdf-getdata cteq61.LHgrid
lhapdf-getdata cteq61.LHpdf
lhapdf-getdata cteq6ll.LHpdf
lhapdf-getdata MRST2004nlo.LHgrid
lhapdf-getdata GSG961.LHgrid
```

HepMC (OSX 10.6.7)

- http://lcgapp.cern.ch/project/simu/HepMC/download/
- downloaded production release i386-mac106-gcc42-opt
- installation directory: `/Users/sevilla/software/hepmc/hepmc-2.06.04`

Whizard installation (OSX 10.6.7)

Whizard 2.0.5

- install gfortran compiler: http://gcc.gnu.org/wiki/GFortranBinaries
- install Objective Caml: http://caml.inria.fr/download.en.html
    - if building from source:
        - `./configure`
        - `make world`
        - `make bootstrap`
        - `make opt`
        - `make opt.opt`
        - `umask 022`
        - `make install`
- download the source tarball: http://projects.hepforge.org/whizard/
- run configure: `./configure --prefix=/Users/sevilla/software/whizard/whizard-2.0.5 FC=/sw/bin/gfortran F77=/sw/bin/gfortran CFLAGS="-g -O2 -m32"`
- do `make` followed by `make install`.

if O'CAML, LHAPDF are installed in non-standard places, need to specify the appropriate env variables. Example:

```
export OCAML=/home/sevilla/software/ocaml/ocaml-3.12.1-i686
export PATH=${OCAML}/bin:${PATH}
export LHAPDF_LIB=/afs/cern.ch/user/s/sevilla/software/lhapdf/lhapdf-5.8.8/lib
```

CompHEP

- After registering, download comphep-4.5.1.tgz from http://comphep.sinp.msu.ru/download/main
- untar into `/Users/sevilla/software/comphep/comphep-4.5.1`
- `./configure --with-gcc4`
    - could not compile with option `--with-lhapdf` nor `--with-root`, so remove the flags...
- change contents of file `CC` from `gcc` to `gcc-4`
- change contents of file `FORLIBS` from `-lgfortran -lgcc` to `-L/sw/lib/gcc4.5/lib -lgfortran -lgcc`
- edit `/Users/sevilla/software/comphep/comphep-4.5.1/Makefile` and add at the beginning `CXX=g++-4`
- `make`
- if the building went ok, one should see at the end:

```
*****************************************************************
* Binaries for CompHEP-4.5.1 has been successfully prepared *
*                                                               *
*      Create a user working directory using the command        *
*         make setup WDIR=path_to_your_user_work_dir            *
*  Note 1: Do not use '~' to refer to you home directory        *
*          Use the environment variable HOME                    *
*****************************************************************
```

- `make setup`
- cd `~/software/comphep/comphep_4.5.1_test`
- `./comphep`

Generators specifics

Herwig

pdgId

The identity codes IDHEP are almost as recommended by the LEP Working Group, i.e. the revised PDG numbers where defined, the only exception is our use of IDHEP = 26 for the lightest MSSM Higgs boson, to distinguish it from the SM Higgs boson (PDG code 25). In addition, the `generator-specific' (IDHEP=81-100) codes 98 and 99 are used for remnant photons and nucleons, respectively.

- IDHEP codes
  - `0` : objects with no PDG code
  - `91` : clusters
  - `94` : jets

Status codes

- 0 : an empty entry, with no meaningful information and therefore to be skipped unconditionally
- 1 : a final-state particle, i.e. a particle that is not decayed further by the generator (may also include unstable particles that are to be decayed later, as part of the detector simulation). Such particles must always be labelled '1'.
- 2: a decayed Standard Model hadron or tau or mu lepton, excepting virtual intermediate states thereof (i.e. the particle must undergo a normal decay, not e.g. a shower branching). Such particles must always be labelled '2'. No other particles can be labelled '2'.
- 3 : a documentation entry
- 4 : an incoming beam particle
- 11 - 200 : an intermediate (decayed/branched/...) particle that does not fulfill the criteria of status code 2, with a generator-dependent classification of its nature

- additional info:
  - http://webber.home.cern.ch/webber/hw65_manual.html#htoc96
  - Herwig++ FAQ

1	final state particle
2	parton before hadronization
3	documentation line
100	cone limiting jet evolution
101	`beam' (beam 1)
102	`target' (beam 2)
103	overall centre of mass
110	unprocessed hard process c.m.
111	unprocessed beam parton
112	unprocessed target parton
113	unproc. first outgoing parton
114	unproc. other outgoing parton
115	unprocessed spectator parton
120-25	as 110-15, after processing
130	lepton in jet (unboosted)
131-34	as 141-44, unboosted to c.m.
135	spacelike parton (beam, unboosted)

136	spacelike parton (target,unboosted)
137	spectator (beam, unboosted)
138	spectator (target, unboosted)
139	parton from branching (unboosted)
140	parton from gluon splitting (unboosted)
141-44	jet from parton type 111-14
145-50	as 135-40 boosted, unclustered
151	as 159, not yet clustered
152	as 160, not yet clustered
153	spectator from beam
154	spectator from target
155	heavy quark before decay
156	spectator before heavy decay
157	parton from QCD branching
158	parton from gluon splitting
159	parton from cluster splitting
160	spectator after heavy decay
161	beam spectator after gluon splitting
162	target spectator after gluon splitting
163	other cluster before soft process
164	beam cluster before soft process
165	target cluster before soft process
167	unhadronized beam cluster
168	unhadronized target cluster
170	soft process centre of mass
171	soft cluster (beam, unhadronized)
172	soft cluster (target, unhadronized)
173	soft cluster (other, unhadronized)
181	beam cluster (no soft process)
182	target cluster (no soft process)
183	hard process cluster (hadronized)
184	soft cluster (beam, hadronized)
185	soft cluster (target, hadronized)
186	soft cluster (other, hadronized)
190-93	as 195-98, before decays
195	direct unstable non-hadron
196	direct unstable hadron (1-body clus.)
197	direct unstable hadron (2-body clus.)
198	indirect unstable hadron or lepton
199	decayed heavy flavour hadron
200	neutral B meson, flavour at prod'n

D3PD variables

- D3PDContentMcTruthBlock
  - `mc_parents`: barcode of parents
  - `mc_children`: barcode of children
  - `mc_parents_index`: index of parents
  - `mc_child_index`: index of children

-- SergioGonzalez - 25-Nov-2011

This topic: Main > McSoftware
Topic revision: r10 - 2018-02-23 - SergioGonzalez