

HepMC

Status & Proposals

Lynn Garren
July 8, 2009

Current Releases

— [HepMC 2.05.00

— released June 5, 2009

— [HepMC 2.04.02

— released June 5, 2009

— [HepMC 2.03.11 – production release

— released July 3, 2009

Recent Changes

— [Units introduced in 2.04

— [IO_Ascii removed in favor of IO_GenEvent in 2.05

— [clarification of status codes

— [streaming I/O operators added in 2.05

— [GenCrossSection added in 2.05

— [I/O rewritten to deal gracefully with corrupt data (2.05)

pyhepmc

— [Andy Buckley has provided a python interface using SWIG

— <http://pypi.python.org/pypi/pyhepmc/0.3.3>

— [this will be maintained as a separate package, outside of HepMC

— [SWIG does have some limitations

Possible Issue

— [Methods modified as of 2.04 to return a const reference instead of a copy

— [ClassX method() const; ==> const ClassX & method() const;

— [affected methods

— GenParticle::flow()

— GenParticle::polarization()

— GenEvent::random_states()

— [Not completely backwards compatible

HepMC 2.06 Discussion

— [the idea is to begin discussion

— [final discussion and decisions in January

— [could move up the schedule if experiments need something

— seems unlikely

HepMC 2.06

— [one hopes for smaller changes

— but perhaps not yet

— already have 7 requests

— [proposed changes listed in random order

Proposal 1

— [Remove deprecated ParticleData, ParticleDataTable, and IO_PDG_ParticleDataTable

— [affects IO_BaseClass

— write_particle_data_table()

— fill_particle_data_table()

— write_particle_data_table()

Proposal 2

— [Have a look at the Pythia and Herwig wrapper functions

— [Is there cleanup that can be done?

— [Do we need to keep wrappers for Pythia 5.72 and 6.152?

— [Since the Fortran versions are essentially frozen, could there be a single Herwig wrapper and a single Pythia wrapper?

Proposal 3a

— [Ascii output has lines for HeavyIon and PdfInfo whether or not they are present in the GenEvent

— [Ascii output only writes a GenCrossSection line if GenCrossSection is present in the GenEvent

— [change HeavyIon and PdfInfo so these lines are only written if present in the event – will require some backporting

— [Comment: this works in the event header, but the vertex and particle information requires a strict format

Proposal 3b

— [could random lines be inserted and ignored?

— all data lines begin with a one character key

— this idea might work

— but what is the rationale?

— would it lead to messy Ascii files?

Proposal 4

current method of iterating over descendants

```
HepMC::GenEvent* ge = ...;  
for (HepMC::GenEvent::particle_const_iterator pi = ge->particles_begin(); pi != ge->particles_end(); ++pi) {  
    (*pi)->pdg_id();  
}
```

current method of iterating over particles

```
HepMC::GenParticle* gp = ...;  
HepMC::GenVertex* gv = gp->end_vertex();  
if (gv) {  
    for (HepMC::GenVertex::particles_out_const_iterator pi = gv->particles_out_const_begin();  
        pi != gv->particles_out_const_end(); ++pi) {  
        (*pi)->pdg_id()  
    }  
}
```

Proposal 4

— [Can this be made more convenient?

— [Add container like access to particles

— `vector<GenParticle*> GenEvent::particles()`

— `vector<const GenParticle*> GenEvent::particles() const`

— `vector<GenVertex*> GenEvent::vertices()`

— `vector<const GenVertex*> GenEvent::vertices() const`

Proposal 4

vector<GenParticle*> GenVertex::particles_in()

vector<const GenParticle*> GenVertex::particles_in() const

vector<GenParticle*> GenVertex::particles_out()

vector<const GenParticle*> GenVertex::particles_out() const

vector<GenParticle*> GenVertex::particles(IteratorRange=relatives)

vector<const GenParticle*> GenVertex::particles(IteratorRange=relatives)
const

Proposal 4

— [example code:

```
foreach (const GenParticle* p, gv.particles(descendants)) {
```

```
    p->pdg_id();
```

```
    ...
```

— [implies possibly removing existing iterators?

— [possible performance hit

Proposal 5

- [Reduce size of tarball

- smaller event dumps for the tests

- [Who does this affect and why?

Proposal 6

- [Change installed location of examples

- [Current location based on CLHEP and user feedback

- examples/HepMC

- allows mixing of examples from different packages

- [use share/HepMC/examples instead?

- better for installation in /usr/local

- [Most builds are on shared AFS or NFS directories, not in /usr

Proposal 7

— [GenEvent::weights() is used in some cases to contain different weights for different PDF's, etc.

— [no option to name these different weights

— [change weights from vector to map

— std::vector<double> becomes std::map<std::string,double>

— GenEvent::weights() returns calculated vector

— GenEvent::named_weights() returns map

— default names?

Final Thoughts

— [growing gap between experiments and other users

— could be a problem