

To test some Event Generator, for example, *QELEventGenerator*:

1. Create an event generator list in *EventGeneratorListAssembler.xml* consisted only a generator under test and a Norm generator, how it's already done for CCQE event generator at *EventGeneratorListAssembler.xml*:

```
<param_set name="CCQE+NORM">
  <param type="int" name="NGenerators"> 2 </param>
  <param type="alg" name="Generator-0"> genie::EventGenerator/QEL-CC </param>
  <param type="alg" name="Generator-1"> genie::EventGenerator/NORM </param>
</param_set>
```

2. Create a dedicated tune with parameters and models configuration for generator under test (see tune EX00_01a in attached archive).

3. Pre-calculate splines for this event generator list using dedicated tune with command:
gmkspl -p 14 -t 1000060120 -o out.xml -n 500 -e 210 --tune EX00_01a_00_000 --event-generator-list CCQE+NORM

4. Generate events with uniform flux (see file *0_210_uniform.root* in the attached archive) with this event generator list and the tune:

```
gevgen -n 1000000 -p 14 -t 1000060120 -e 0.01,210 -f 0_210_uniform.root,numu --cross-sections out.xml --tune EX00_01a_00_000 --event-generator-list CCQE+NORM -o Total.root
```

5. Plot the energy distribution for events in *Total.root* generated with the **NormEventGenerator**, for example:

```
// Loop over all events
for(int i = 0; i < nev; i++)
{

  // get next tree entry
  tree->GetEntry(i);
  if (i%10000 == 0)
    std::cout << "Event #" << i << std::endl;

  // get the GENIE event
  EventRecord & event = *(mcrec->event);

  Interaction * interaction = event.Summary();
  InitialState init_state = interaction->InitState();
  ProcessInfo proc_info = interaction->ProcInfo();
  bool isNorm = proc_info.IsNorm();

  double E = init_state.ProbeE(kRfLab);
  if (isNorm)
    hNorm->Fill(E);
}
```

}

6. This distribution by energy (see file *norm.root* in the attached archive) should be flat in the whole interested energy range. If it is not so (how in the figure below), then there are some problems with event generator under test. The explanation of such behavior could be found in the presentation.

