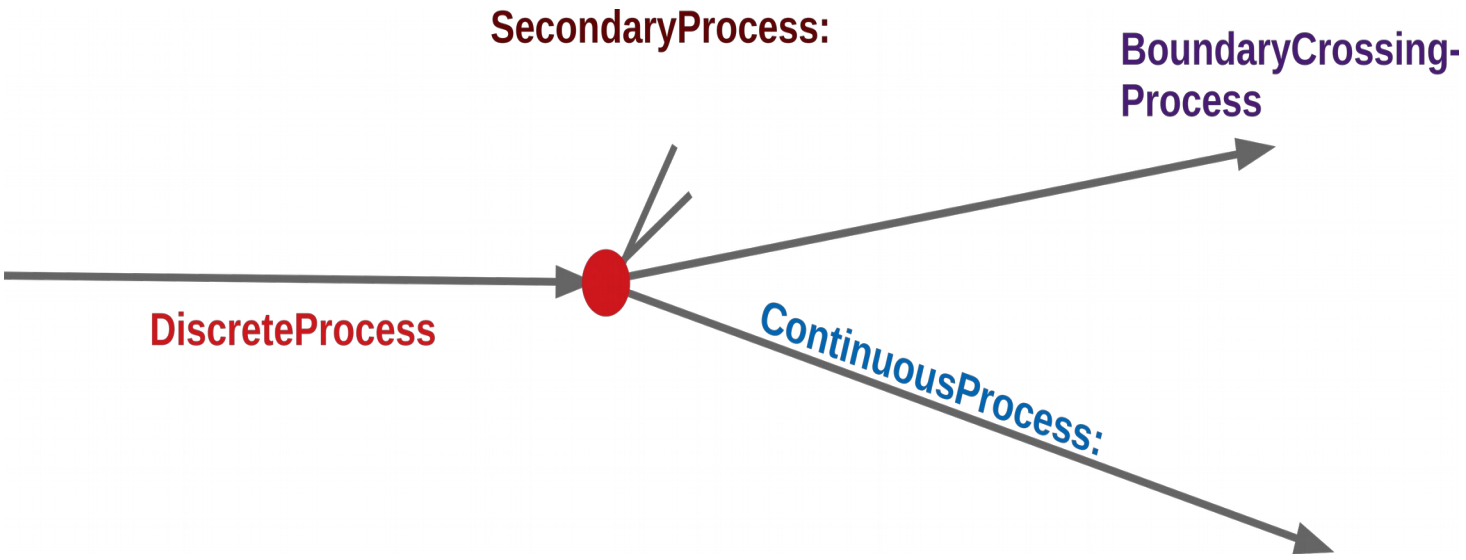
Processes in Corsika 8

Hands-on @ corsika workshop 2019

Classes of processes

Different kinds for different jobs:

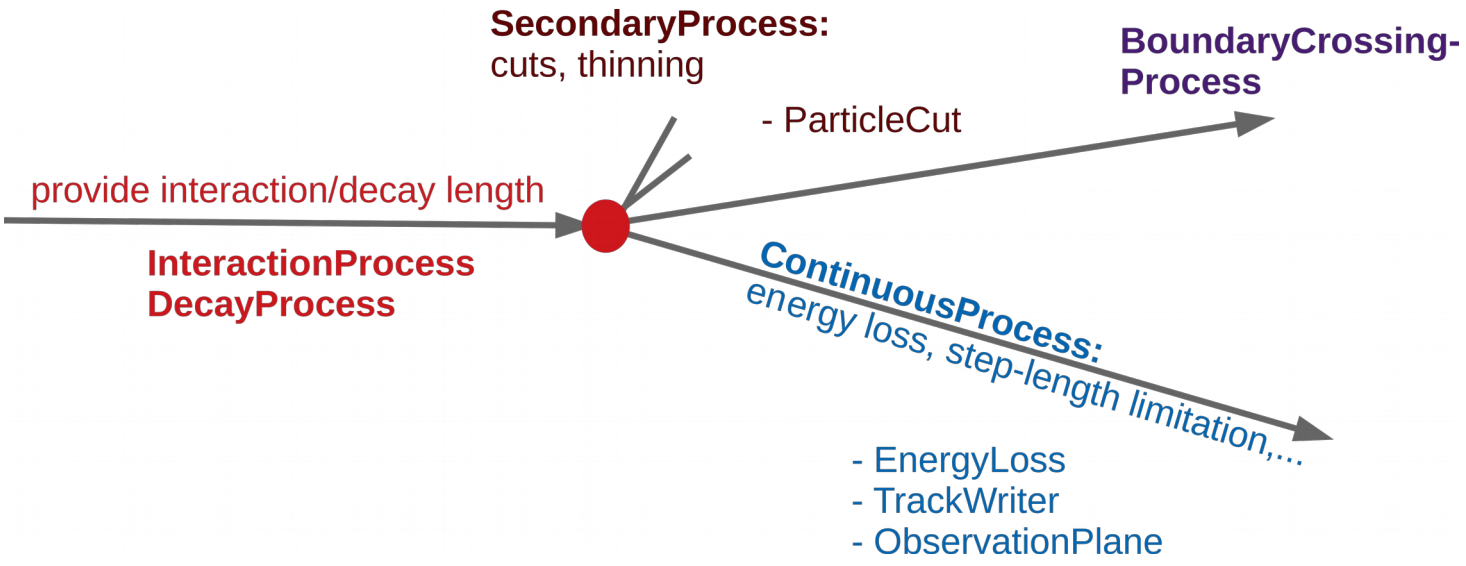
* Stack
* Discrete
* Continous
* Secondaries
* BoundaryCrossing



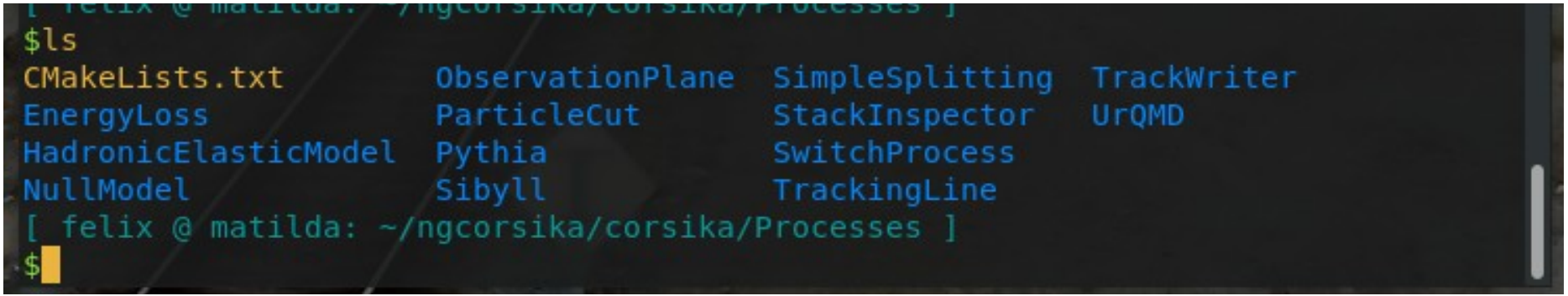
Located in:
Framework/ProcessSequence

Concrete Processes

Located in: Processes/



Lets have a look ...


$ls
CMakeLists.txt
EnergyLoss
HadronicElasticModel
NullModel
ObservationPlane
ParticleCut
Pythia
Sibyll
SimpleSplitting
StackInspector
SwitchProcess
TrackingLine
TrackWriter
UrQMD
[felix @ matilda: ~/ngcorsika/corsika/Processes]
$

Adding a new process

we do need EM cascades, so lets add a splitting a la Heitler

1) create directory in Processes/
2) create a class inheriting from InteractionProcess
    - needs functions: GetInteractionLength and DoInteraction
3) fill with physics ...

a prototype in Framework/Cascade/testCascade.cc

ProcessSplit

InteractionProcess:

DoInteraction:
* takes projectile
* Adds secondaries

```
class ProcessSplit : public process::InteractionProcess<ProcessSplit> {

  int fCalls = 0;
  GrammageType fX0;

public:
  ProcessSplit(GrammageType const X0)
      : fX0(X0) {}

  template <typename Particle>
  corsika::units::si::GrammageType GetInteractionLength(Particle const&) const {
    return fX0;
  }

  template <typename TProjectile>
  corsika::process::EProcessReturn DoInteraction(TProjectile& vP) {
    fCalls++;
    const HEPEnergyType E = vP.GetEnergy();
    vP.AddSecondary(
        std::tuple<particles::Code, units::si::HEPEnergyType,
                   corsika::stack::MomentumVector, geometry::Point, units::si::TimeType>{
            vP.GetPID(), E / 2, vP.GetMomentum(), vP.GetPosition(), vP.GetTime()});
    vP.AddSecondary(
        std::tuple<particles::Code, units::si::HEPEnergyType,
                   corsika::stack::MomentumVector, geometry::Point, units::si::TimeType>{
            vP.GetPID(), E / 2, vP.GetMomentum(), vP.GetPosition(), vP.GetTime()});
    return EProcessReturn::eInteracted;
  }

  void Init() { fCalls = 0; }

  int GetCalls() const { return fCalls; }
};
```

ProcessCut

SecondariesProcess:

DoSecondaries:

* takes limited StackIterator
* can modify or delete secondary particles !!

```
class ProcessCut : public process::SecondariesProcess<ProcessCut> {

  int fCount = 0;
  int fCalls = 0;
  HEPEnergyType fEcrit;

public:
  ProcessCut(HEPEnergyType e)
      : fEcrit(e) {}

  template <typename TStack>
  EProcessReturn DoSecondaries(TStack& vS) {
    fCalls++;
    auto p = vS.begin();
    while (p != vS.end()) {
      HEPEnergyType E = p.GetEnergy();
      if (E < fEcrit) {
        p.Delete();
        fCount++;
      } else {
        ++p; // next particle
      }
    }
    cout << "ProcessCut::DoSecondaries size=" << vS.GetSize() << " count=" << fCount
         << endl;
    return EProcessReturn::eOk;
  }

  void Init() {
    fCalls = 0;
    fCount = 0;
  }

  int GetCount() const { return fCount; }
  int GetCalls() const { return fCalls; }
};
```

Assemble cascade

```
HEPEnergyType E0 = 100_GeV;

random::RNGManager& rmng = random::RNGManager::GetInstance();
rmng.RegisterRandomStream("cascade");

auto env = MakeDummyEnv();
tracking_line::TrackingLine tracking;

stack_inspector::StackInspector<TestCascadeStack> stackInspect(1, true, E0);
null_model::NullModel nullModel;

const GrammageType X0 = 20_g / square(1_cm);
const HEPEnergyType Ecrit = 85_MeV;
ProcessSplit split(X0);
ProcessCut cut(Ecrit);
auto sequence = nullModel << stackInspect << split << cut;
TestCascadeStack stack;

cascade::Cascade<tracking_line::TrackingLine, decltype(sequence), TestCascadeStack,
                 TestCascadeStackView>
    EAS(env, tracking, sequence, stack);
CoordinateSystem const& rootCS =
    RootCoordinateSystem::GetInstance().GetRootCoordinateSystem();

stack.Clear();
stack.AddParticle(
    std::tuple<particles::Code, units::si::HEPEnergyType,
               corsika::stack::MomentumVector, geometry::Point, units::si::TimeType>{
        particles::Code::Electron, E0,
        corsika::stack::MomentumVector(rootCS, {0_GeV, 0_GeV, -1_GeV}),
        Point(rootCS, {0_m, 0_m, 10_km}), 0_ns});
EAS.Init();
EAS.Run();
```

Fun & easy tasks

add neutron decay
#129 · opened 4 months ago by Felix Riehn Full physics demonstratror Development Feature request

Add pion decay routine
#110 · opened 5 months ago by Ralf Ulrich First full release: CORSIKA 8.0.0 Feature request Critical

add muon decay
#108 · opened 5 months ago by Felix Riehn Full physics demonstratror Feature request Important

+ Hillas splitting algorithm

+ NKG treatment of EM