

## Decays of “Stable” Tracks in fast-sim

Four classes created to determine where “stable” particles decay in detector and create their daughters.

- **PacDKPointInfo** - contains decay point information
- **PacDKPointGenerator** - creates PacDKPointInfo objects
- **PacDKChainInfo** - contains decay chain information
- **PacDKChainGenerator** - creates PacDKChainInfo objects

## Some code from testChargedTrackDK.cc

```
// instantiate a PacDKPointInfo for passing decay point
// information between the PacDKPointGenerator object and
// this code. Also, a PacDKChainInfo and a PacDKChainGenerator.
PacDKPointInfo* theTrackDecay = new PacDKPointInfo();
PacDKPointGenerator* dkPointGen = new PacDKPointGenerator();
PacDKChainGenerator* dkChainGen = new PacDKChainGenerator();
PacDKChainInfo* dkChainInfo;

...

//Simulate Track through detectors
PacSimTrack* simtrk = sim.simulateGTrack(&gtrk);

// find out if/where simtrk decays
dkPointGen->generateDecayPoint(simtrk, theTrackDecay);

PacDKPointInfo* theSecondTrackDecay =
    dkPointGen->generateDecayPoint(simtrk);
```

## More code from testChargedTrackDK.cc

```
// find the decay products if the track decays in the detector
// the code should assert if the PacSimTrack does not decay in detector.
if (theTrackDecay->decaysInDetector()) {
    // dkChainInfo was "declared" earlier so that it will stay in scope
    // outside this "if (..) { ... }"
    dkChainInfo = dkChainGen->generateDecayChain(theTrackDecay);
} // end of "if (theTrackDecay->decaysInDetector)" ...

const PacSimTrack* theTrackDecaySimtrk=
    theTrackDecay->getPacSimTrack();
const GTrack* theDecayGTrack = theTrackDecaySimtrk->getGTrack();

if (theTrackDecay->decaysInDetector()) {
    cout << theTrackDecay->decayHitIndex() << endl;
    cout << theTrackDecay->decayPoint() << endl;
    cout << theTrackDecay->decayDirection() << endl;
    cout << theTrackDecay->decayPointMom() << endl;
```

## Code looking at PacDKChainInfo content

```
const std::vector<EvtParticle *>& theDaughters
    = dkChainInfo->getDaughters();
for(int i = 0; i < theDaughters.size(); i++) {
    cout << endl << "information about aDaughter" << endl;
    EvtParticle* aDaughter = theDaughters[i];
    aDaughter->printParticle();
    cout << "aDaughter->getId() = " << aDaughter->getId() << endl;
    std::string aDaughterName = EvtPDL::name(aDaughter->getId());
    cout << "aDaughterName = " << aDaughterName << endl;
    int aDaughterStdHep = EvtPDL::getStdHep(aDaughter->getId());
    cout << "aDaughterStdHep = " << aDaughterStdHep << endl;
    EvtVector4R aDaughterP4 = aDaughter->getP4Lab();
    cout << " aDaughterP4 = " << aDaughterP4 << endl;
} // end of "for (int i =0, ..."
```

# The DK.DEC file

Cloned DECAY.DEC file and added branching fractions for  $K^+$ ,  $K^-$ ,  $\pi^+$ ,  $\pi^-$ ,  $\mu^+$ , and  $\mu^-$  decays. Have not yet added branching fractions for  $K^0_L$  decays.  $K^0_{short}$ ,  $\Lambda$ ,  $\Sigma$  decays already found in DECAY.DEC.

```
# add some K+ decay modes 081022 - mds
```

```
Decay K+
```

0.6343	$\mu^+ \nu_\mu$	SLN;
0.0487	$\pi^0 e^+ \nu_e$	PHSP;
0.0327	$\pi^0 \mu^+ \nu_\mu$	PHSP;
0.2113	$\pi^+ \pi^0$	PHSP;
0.0173	$\pi^+ \pi^0 \pi^0$	PHSP;
0.5576	$\pi^+ \pi^+ \pi^-$	PHSP;

```
Enddecay
```

## Summary of Status

- Code is useable, as is.
- Need to add Klong decay modes to DK.DEC
- Rolf and Dave Brown have made suggestions for improving design/implementation of code.
- Dave Brown has suggested creating a PacDecayInfo class where each object would include both a PacDKPointInfo and a PacDKChainInfo (and other information as well).
- Will modify design of classes (may drop some data members, accessor methods, etc.) after we have determined how the PacDKPointInfo and PacDKChainInfo objects will be used.