L3 Muon Structure

Flow Process

hltL3TrajSeedOIState
hltL3TrackCandidateFromL2OIState
hltL3TkTracksFromL2OIState
hltL3MuonsOIState
hltL3TrajSeedOIHit
hltL3TrackCandidateFromL2OIHit
hltL3TkTracksFromL2OIHit
hltL3MuonsOIHit
hltL3TkFromL2OICombination
hltL3TrajSeedIOHit
hltL3TrackCandidateFromL2IOHit
hltL3TkTracksFromL2IOHit
hltL3MuonsIOHit
hltL3TrajectorySeed
hltL3TrackCandidateFromL2

After the local pixel and strip sequences the three different seeding algorithms are visited in the sequence: OIState, OIHit and IOHit.

The steps for each algorithm is described below starting with the creation of L3 muon trajectory seeds from the hltL3TrajSeed.

The following notation is used:

class::function() (products) [inheritance]

---

hltL3TrajSeedOIState

TSGFromL2Muon::produce() (L3MuonTrajectorySeedCollection) [edm::EDProducer]

MuonTrackingRegionBuilder::setEvent()

TrackerSeedCleaner::setEvent()

MuonTrackingRegionBuilder::region()

TSGForRoadSearch::trackerSeeds()

TrackerSeedCleaner::clean()

hltL3TrajSeedOIHit and hltL3TrajSeedIOHit

TSGFromL2Muon::produce() (L3MuonTrajectorySeedCollection) [edm::EDProducer]

MuonTrackingRegionBuilder::setEvent()

TrackerSeedCleaner::setEvent()

MuonTrackingRegionBuilder::region()

SeparatingTSG::trackerSeeds()

DualByL2TSG::selectTSG() // Here we choose whether or not to use this seeding algorithm

TrackerSeedCleaner::clean()

hltL3TrackCandidateFromL2

CkfTrackCandidateMakerBase::produceBase() (std::vector<Trajectory>,TrackCandidateCollection)

cms::CkfTrackCandidateMakerBase::setEventSetup()

NavigationSetter::NavigationSetter()

BaseCkfTrajectoryBuilder::clone() // A: create Trajectory builder

RedundantSeedCleaner::init() // D: invoke building algo

BaseCkfTrajectoryBuilder::buildTrajectories() // Build trajectory from seed outwards

TrajectoryCleaner::clean() // select best trajectory

BaseCkfTrajectoryBuilder::rebuildTrajectories() // possibly continue building trajs back thru seed

TrajectoryCleaner::clean() // select best trajectory

RedundantSeedCleaner::done()

TrajectoryCleaner::clean() // E: clean results to avoid dup tracks

// If requested, reverse the trajectories creating a new 1-hit seed on the last measurement of the track

Trajectory::recHitsV() // F: Convert to TrackCandidates

TransientInitialStateEstimator::innerState()

thePropagator->propagate()

trajectoryStateTransform::persistentState()

hltL3TkTracksFromL2

TrackProducer::produce() (TrackingRecHitCollection,reco::TrackCollection,reco::TrackExtraCollection,std::vector<Trajectory>) [KfTrackProducerBase, edm::EDProducer]

TrackProducerBase< reco::Track >::getFromES()

TrackProducerBase< reco::Track >::getFromEvt()

TrackProducerAlgorithm::runWithCandidate()

hltL3Muons

L3MuonProducer::produce() () [edm::EDProducer]

MuonServiceProxy::update()

MuonTrajectoryBuilder::TrackCand()

MuonTrackFinder::reconstruct()

L3MuonTrajectoryBuilder::trajectories() //reconstruct trajectories from standalone and tracker only Tracks -> turn tkMatchedTracks into MuonCandidates

GlobalMuonTrackMatcher::match()

MuonCandidate::MuonCandidate()

GlobalTrajectoryBuilderBase::build()

MuonTrajectoryCleaner::clean()

MuonTrackFinder::load(TrajectoryContainer,edm::Event) //returns reco::TrackCollection

MuonTrackLoader::loadTracks(trajectories, event) //Convert the trajectories into tracks and load the tracks in the event