Table of Contents

Data Format Tables for the Offline Guide

Complete:

This is a page to document the data formats in the Offline Guide. This page has replaced the information in the Reference Manual for RECO and AOD

Separate RECO, AOD and SIM tables

The tables can be found in the following two pages:

- SWGuideRecoDataTable for RECO (including AOD)
- SWGuideAodDataTable for AOD
- SWGuideSimDataTable for SIM in RECOSIM and AODSIM (in construction)

Data Format pages

The full table is put together from separate pages corresponding to EventContent_cff.py. These pages are for RECO and AOD

- SWGuideDataFormatRecoTracker, responsible Giuseppe Cerati, Giovanni Petrucciani
- SWGuideDataFormatRecoLocalTracker, responsible Urs Langenegger, Hella Snoek (PixelTracker), Michael Segala, Rebeca Gonzalez (StripTracker)
- SWGuideDataFormatRecoLocalCalo, responsible ?
- SWGuideDataFormatRecoEcal, responsible Stefano Argiro
- SWGuideDataFormatRecoEgamma, responsible Matteo Sani
- SWGuideDataFormatRecoLocalMuon, responsible Tim Cox (CSC), Mario Pelliccioni (DT), Camilo Carrillo (RPC)
- SWGuideDataFormatRecoMuon, responsible Vyacheslav Krutelyov
- SWGuideDataFormatRecoJets, responsible Roman Kogler
- SWGuideDataFormatRecoMET, responsible Tai Sakuma
- SWGuideDataFormatRecoBTag, responsible Tommaso Boccali
- SWGuideDataFormatRecoTauTag, responsible Simone Gennai, Evan Friis
- SWGuideDataFormatRecoVertex, responsible ?
- SWGuideDataFormatRecoPixelVertexing, responsible ?
- SWGuideDataFormatHLTrigger, responsible Christos Leonidopoulos
- SWGuideDataFormatRecoParticleFlow, responsible Colin Bernet

for SIM (new):

- SWGuideDataFormatGeneratorInterface, responsible
- SWGuideDataFormatSimG4Core, responsible
- SWGuideDataFormatSimTracker, responsible
- SWGuideDataFormatSimMuon, responsible
- SWGuideDataFormatSimCalorimetry, responsible ... (not concerned: no RECOSIM or AODSIM)
- SWGuideDataFormatSimGeneral, responsible ... (not concerned: no RECOSIM or AODSIM)
- SWGuideDataFormatIOMC, responsible ... (not concerned: no RECOSIM or AODSIM)

for cosmic data (cosmics specific event constents from EventContentCosmics_cff.py)

- SWGuideDataFormatRecoMuonCosmics in construction
- SWGuideDataFormatRecoLocalMuonCosmics in construction
- SWGuideDataFormatRecoEcalCosmics in construction

- SWGuideDataFormatRecoLocalCaloCosmics in construction
- SWGuideDataFormatRecoLocalTrackerCosmics in construction
- SWGuideDataFormatL1TriggerCosmics

There is one-to-one correspondance of the pages and people responsible to keep them up-to-date. Tommaso has provided the names and I have contacted these persons on June 28th (see the original mail, the first and the second reminder, and the final check).

No permission to view CMS.RecoContacts

The table should provide the necessary information to access the data. It should be clear where an entry from this table should be added to the skeleton provided by mkedanlzr. Together with the table, the following example is provided:

How to use the table

In your analyzer, you should have the header of the data format you will access - you find this by clicking on the entry in the **Containers** column - and in you should add the corresponding entries in the BuildFile, see WorkBookBuildFilesIntro.

Do the following in `analyze` method of your analyzer

```
  Handle<reco::TrackCollection> tracks;
  iEvent.getByLabel("generalTracks",tracks);
  for(reco::TrackCollection::const_iterator itTrack = tracks->begin();
    itTrack != tracks->end();
    ++itTrack) {
      //  your code
}
```

where `reco::TrackCollection` comes from the **Containers** column and `generalTracks` from the **InputTag/Module** column in the following table. If the data you want to access has an instance name (in parenthesis in the first column) you should use the corresponding `getByLabel` method:

```
  Handle<reco::SuperCluster> clusters;
  iEvent.getByLabel("islandSuperClusters","islandEndcapSuperClusters", clusters);
  for(reco::SuperCluster::const_iterator itClust = clusters->begin();
    itClust != clusters->end();
    ++itClust) {
      //  your code
}
```

Note that for the sake of commodity, you can define the input tag name (and instance names) in the configuration file so you do not have to recompile your code when you want to access objects of different types. This is already done for you if you use the mkedanlz script with -track option and in many tutorials in the WorkBook. The basic use of the script is explained in WorkBookWriteFrameworkModule.

Instructions to complete the table

I have included some basic information based on the corresponding event content files. This is obviously very, very incomplete due to the wild cards. Note that

- the data formats are documented in pages listed above, and you should complete the information as in 3_1_X (in case of differences between the preivous versions see the note below)
  - ♦ follow the link to "your" page
  - ♦ start editing taking care that you add the "both RECO and AOD" data and "RECO only" data

in the correct parts, respectively (green and blue)

- all the entries which you get from AOD and RECO when running `module dump = EventContentAnalyzer {}` should be documented (for your convenience, these are provided in RECO output (updated for 2_0_0) and for an AOD output (updated for 2_0_x), please check that the table documents all entries in these files).
- for **cosmics**, there are separate tables
  - find the event dump in cosmics output (for 3_1_X)
  - if convenient, fill in to the cosmic specific pages (SWGuideDataFormatReco*Cosmics) only the formats which differ from the standard collision running, add the link to the "standard" page (SWGuideDataFormatReco*) for the rest
- the first column gives the module name (the instance name in parenthesis if needed), it should be in the exact form in which it will be entered to getByLabel method (see the instructions above)
- the second column gives the collection name in the exact form in which it entered in Handle in the instructions above
- the link from the second column points to the class documentation in the reference manual, use the format given as an example, i.e.

```
[[%DOXY%r=%LATEST21X%&c=Track&K={reco}][reco::TrackCollection]]
```

  (see details). This points to the header which needs to be added in the analyzer.
- do not change the format of the table.

The different versions will be presented separately as in SWGuideDataFormatRecoBTag.

The complete table

And now to the table itself. Note that table fractions are included from the pages where they are defined with INCLUDE statements. A complete table with RECO and AOD would look like

```
<!--STARTTitleDF-->
%TABLE{ tablewidth="100%" columnwidths="30%,30%,40%"}%
| *InputTag/Module (Instance name)* | *Containers* | *Description* |
<!--STOPTitleDF-->
%INCLUDE{"SWGuideDataFormatRecoTracker" pattern="(?:.*?<!--STOPRecoTrackerDF-->){0}.*?<!--STARTRe
etc
```

and it results in:

InputTag/Module (Instance name)	Containers	Description
Track collections (in RECO and AOD)		
generalTracks	reco::TrackCollection	Collection of tracks obtained with tracker-standalone reconstruction and officially supported by the Tracker DPG group. Such a collection can contain tracks from different tracking algorithms
Track collections (in RECO only)		
generalTracks	reco::TrackExtraCollection	Track extra for the generalTracks.The trajectory state at the inner and outer most measurements

Local calorimetry reco collections (in RECO only)		
hbhereco	edm::SortedCollection<HBHERecHit>	Joint HCAL barrel+endcap RecHits collection

hfreco	edm::SortedCollection<HFRecHit>	Very Forward calorimeter RecHits collection
horeco	edm::SortedCollection<HORecHit>	Outer clorimeter RecHits collection
zdcreco	edm::SortedCollection<ZDCRecHit>	Zero-degree calorimeter RecHits collection
castorreco	edm::SortedCollection<CastorRecHit>	Collection of CastorRecHits containing energy deposits for all channels
ecalRecHit,EcalRecHitsEB	edm::SortedCollection<EcalRecHit>	Collection of Ecal Hits in EB
ecalRecHit,EcalRecHitsEE	edm::SortedCollection<EcalRecHit>	Collection of Ecal Hits in EE
ecalPreshowerRecHit,EcalRecHitsES	edm::SortedCollection<EcalRecHit>	Collection of Ecal Hits in ES

ECAL cluster collections (in RECO and AOD)		
reducedEcalRecHitsEB:reducedEcalRecHitsEB ditto for EE,ES	edm::SortedCollection	Rechits from a 5x5 around Basic Clusters, for the ES, hits corresponding to clusters in EE
hybridSuperClusters	reco::BasicClusterCollection reco::ClusterShapeCollection reco::BasicClusterShapeAssociationCollection reco::SuperClusterCollection	Basic clusters, cluster shapes and super-clusters reconstructed with the hybrid algorithm with no energy corrections applied (barrel only)
hybridSuperClusters:uncleanOnlyHybridSuperClusters	reco::SuperClusterCollection	Only the SuperClusters containing anomalous signals, with no cleaning
correctedHybridSuperClusters	reco::SuperClusterCollection	Super-clusters reconstructed with the hybrid algorithm with energy corrections applied (barrel only)
multi5x5BasicClusters::multi5x5EndcapBasicClusters	reco::BasicClusterCollection	Basic clusters in EE, 5x5 algorithm
correctedMulti5x5SuperClustersWithPreshower	reco::SuperCluster	Corrected superclusters

		in EE, 5x5 algorithm
multi5x5SuperClustersWithPreshower	reco::PreShowerClusters	Preshower clusters

Egamma electron collections (in RECO and AOD)		
gsfElectronCores	reco::GsfElectronCoreCollection	Offline electron object containing a minimal set of information: reference to GsfTrack, reference to SuperCluster (Egamma and/or PFlow cluster) and provenance information. The collection is unpreselected and contains electrons reconstructed either by Egamma or by PF algorithm.
gsfElectrons	reco::GsfElectronCollection	Offline electron containing all the relevant observables and corrections. The collection is produced starting from GsfElectronCore and its modularity allows to easily recompute electron variables skipping the reconstruction step. Each electron has also a reference to the corresponding "core" partner. (**standard offline electron collection**)
electronGsfTracks	reco::GsfTrackCollection	GsfTracks for GsfElectrons (**produced in a dedicated module in the standard reconstruction sequence**). They have been produced from the electronMergedSeeds described below.
eidTYPE		ValueMaps containing a reference to a CMS.GsfElectron and the result of the electron ID. The event has one Value Map per identification TYPE (eidLoose, eidRobust...) (**standard electron ID computed in standard sequence**)
siStripElectronToTrackAssociator	reco::ElectronCollection	Electrons reconstructed using tracks seeded in the Si strip layers (**not produced in standard sequence**)
egammaCTFFinalFitWithMaterial	reco::TrackCollection	Tracks for Si strip seeded electrons (**not produced in standard sequence**)
siStripElectrons	reco::SiStripElectronCollection	Intermediate object for Si Strip electron reconstruction containing information about Si strip hits associated to the SuperCluster (**not produced in standard sequence**)
Egamma photon collections (in RECO and AOD)		
photonCore	reco::PhotonCoreCollection	Photon objects containing reference to SuperCluster, Conversions and ElectronSeeds.
photons	reco::PhotonCollection	Photons with all the related

		observables like vertex, shower shapes, isolation. Each photon has also a reference to the corresponding "core" partner. (**standard reconstructed photon collection**)
PhotonIDProd	reco::PhotonIDCollection	Photons identification variables, calculated for corresponding photons. (**standard reconstructed photon collection**)
conversions	reco::ConversionCollection	Converted photons
ckfOutInTracksFromConversions	reco::TrackCollection	Conversion tracks from outside-in tracking.
ckfInOutTracksFromConversions	reco::TrackCollection	Conversion tracks from inside-out tracking.
Egamma electron collections (in RECO only)		
gsfElectronsGsfFit	reco::GsfTrackExtraCollection reco::TrackExtraCollection TrackingRecHitCollection	GsfTrackExtras, TrackExtras, TrackingRecHits
electronMergedSeeds	reco::ElectronSeedCollection	Mixed collection of ElectronSeeds coming from Egamma and/or PFlow algorithms, with their parent SuperClusters and/or tracks. They are used as input for the production of electronGsfTracks.
ecalDrivenElectronSeeds	reco::ElectronSeedCollection	Collection of ElectronSeeds with their parent SuperClusters, using the SuperCluster driven pixel matching algorithm, made for electron track seeding. That's one of the two collections which are mixed into electronMergedSeeds just above.
egammaCTFFinalFitWithMaterial	reco::TrackExtraCollection	TrackExtras for Si strip seeded electrons (**not produced in standard sequence**)
Egamma photon collections (in RECO only)		
ckfOutInTracksFromConversions	reco::TrackExtraCollection TrackingRecHitCollection	TrackExtras and TrackingRecHits for conversion tracks from outside-in tracking.
ckfInOutTracksFromConversions	reco::TrackExtraCollection TrackingRecHitCollection	TrackExtras and TrackingRecHits for conversion tracks from inside-out tracking.

Muon collections (in RECO and AOD)		
muons	reco::MuonCollection	Muons built using track global-muon reconstruc information (energy dep
muonsFromCosmics	reco::MuonCollection	Similar to "muons" but reconstructor (2-leg opt
muonsFromCosmics1Leg	reco::MuonCollection	Similar to "muons" but reconstructor (1-leg opt
Muon collections (in RECO only)		
calomuons	reco::CaloMuonCollection	Tracks with energy dep with those of a muon, b reconstruction algorithr

muonsWithSET	reco::MuonCollection	Similar to "muons" but algorithm for standalon
Muon track collections (in RECO and AOD)		
standAloneMuons	reco::TrackCollection	Standalone muon tracks
standAloneMuons:UpdatedAtVtx	reco::TrackCollection	Standalone muon tracks the beam spot
cosmicMuons	reco::TrackCollection	Standalone muon tracks cosmic-muon reconstru
cosmicMuons1Leg	reco::TrackCollection	Standalone muon tracks cosmic-muon reconstru
globalMuons	reco::TrackCollection	Global muon tracks wit
globalCosmicMuons	reco::TrackCollection	Global muon tracks rec reconstructor (2-leg opt
globalCosmicMuons1Leg	reco::TrackCollection	Global muon tracks rec reconstructor (1-leg opt
tevMuons:default	reco::TrackCollection	Global muon tracks usi algorithm with one refit
tevMuons:firstHit	reco::TrackCollection	Global muon tracks usi muon station that has th
tevMuons:picky	reco::TrackCollection	Global muon tracks usi muon stations which do
tevMuons:dyt	reco::TrackCollection	Global muon tracks rec high-p_T muons
Muon track collections (in RECO only)		
standAloneSETMuons	reco::TrackCollection	Similar to standAloneM algorithm
globalSETMuons	reco::TrackCollection	Similar to globalMuons algorithm
Muon seeds (in RECO only)		
MuonSeed	TrajectorySeed	Seeds for standalone mu
CosmicMuonSeed	TrajectorySeed	Seeds for dedicated cos
Muon isolation collections (in RECO and AOD)		
muIsoDepositTk	reco::IsoDepositMap	Map of IsoDeposits for tracks
muIsoDepositCalByAssociatorTowers	reco::IsoDepositMap	Map of IsoDeposits for towers. Three instances calo-tower component):
muIsoDepositJets	reco::IsoDepositMap	Map of IsoDeposits for (sisCone5CaloJets)
Muon track associations (in RECO and AOD)		
tevMuons:default	reco::TrackToTrackMap	Map associating tracks in the tevMuons:default
tevMuons:firstHit	reco::TrackToTrackMap	Map associating tracks in the tevMuons:firstHi
tevMuons:picky	reco::TrackToTrackMap	Map associating tracks in the tevMuons:picky c
tevMuons:dyt	reco::TrackToTrackMap	Map associating tracks in the tevMuons:dyt col
Other information (in RECO and AOD)		
muid*	edm::ValueMap<boolean>	Output of the muon sele DataFormats/MuonRec

MuonShowerInformation	edm::ValueMap<reco::MuonShower>	Muon shower informati DataFormats/MuonRec
muons	edm::ValueMap<reco::MuonTimeExtra>	Muon timing informatic in DataFormats/MuonR
cosmicsVeto	edm::ValueMap<unsigned int>	Index of the partner trac used by the cosmic-muo
cosmicsVeto	edm::ValueMap<reco::MuonCosmicCompatibility>	Information used by the DataFormats/MuonRec

Particle Flow jet collections (in RECO and AOD)		
ak5PFJets	reco::PFJetCollection	Fastjet Anti-kT R=0.5 jets reconstructed from PF particles.
ak7PFJets	reco::PFJetCollection	Fastjet Anti-kT R=0.7 jets reconstructed from PF particles.
kt4PFJets	reco::PFJetCollection	Fastjet kT R=0.4 jets reconstructed from PF particles
kt6PFJets	reco::PFJetCollection	Fastjet kT R=0.6 jets reconstructed from PF particles
Calo jet collections (in RECO and AOD)		
ak5CaloJets	reco::CaloJetCollection	Fastjet Anti-kT R=0.5 jets reconstructed from CaloTowers with pT>0.5 GeV.
ak7CaloJets	reco::CaloJetCollection	Fastjet Anti-kT R=0.7 jets reconstructed from CaloTowers with pT>0.5 GeV.
kt4CaloJets	reco::CaloJetCollection	Fastjet kT R=0.4 jets reconstructed from CaloTowers with pT>0.5 GeV
kt6CaloJets	reco::CaloJetCollection	Fastjet kT R=0.6 jets reconstructed from CaloTowers with pT>0.5 GeV
caloTowers		
towerMaker		
Basic jet collections (in RECO and AOD)		
ak7BasicJets	reco::CastorTowerCollection	Fastjet Anti-kT R=0.7 jets reconstructed from CastorTowers
ak7CastorJetID	reco::CastorJetIDValueMap	Corresponding JetID object to go with the ak7BasicJets, contains various information on how a jet in CASTOR looks, see CASTOR reconstruction page for more info
CastorTowerReco	reco::CastorTowerCollection	Collection of towers in CASTOR (RecHits in one phi sector summed over z)
Jet-plus-tracks jet collections (in RECO and AOD)		
JetPlusTrackZSPCorJetAntiKt5	reco::JPTJetCollection	Fastjet Anti-kT R=0.5 jets reconstructed from CaloTowers, corrected with track response within the jet cone.
Track jet collections (in RECO and AOD)		
ak5TrackJets	reco::TrackJetCollection	Fastjet Anti-kT R=0.5 jets reconstructed from tracks.
Gen jet collections (in RECO and AOD)		
ak5GenJets	reco::GenJetCollection	Fastjet Anti-kT R=0.5 jets reconstructed from stable generator particles.
ak7GenJets	reco::GenJetCollection	Fastjet Anti-kT R=0.7 jets reconstructed from stable generator particles.
kt4GenJets	reco::GenJetCollection	

		Fastjet kT R=0.4 jets reconstructed from stable generator particles
kt6GenJets	reco::GenJetCollection	Fastjet kT R=0.6 jets reconstructed from stable generator particles
genParticleCandidates		
Jet extension collections (in RECO and AOD) for 3_3_X (FIXME)		
ak5JetTracksAssociatorAtVertex		tracks associated to all ak5CaloJets within a Cone R=0.5 at the vertex

PFMET (in RECO and AOD)		
pfMet	reco::PFMETCollection	MET of all reconstructed particles with the particle flow algorithm
pfChMet	reco::PFMETCollection	MET of charged particles which are reconstructed with the particle flow algorithm whose corresponding track has the dz parameter with respect to the the main vertex smaller than 0.2 cm
GenMET (in RECO and AOD)		
genMetTrue	reco::GenMETCollection	MET of all generated particles in simulation in their final states but excluding neutrinos, excited neutrinos, right-handed neutrinos, sneutrinos, neutralinos, gravitons, gravitinos
genMetCalo	reco::GenMETCollection	MET of all generated particles in simulation in their final states but excluding neutrinos, excited neutrinos, right-handed neutrinos, sneutrinos, neutralinos, gravitons, gravitinos and also muons
genMetCaloAndNonPrompt	reco::GenMETCollection	MET of all generated particles in simulation in their final states but excluding excited neutrinos, right-handed neutrinos, sneutrinos, neutralinos, gravitons, gravitinos and, additionally, excluding muons and neutrinos coming from the decay of gauge bosons and top quarks
CaloMET (in RECO and AOD)		
caloMet	reco::CaloMETCollection	MET of all energy deposits in calorimeter towers in EB, EE, HB, HE, and HF
caloMetBE	reco::CaloMETCollection	MET of all energy deposits in calorimeter towers in EB, EE, HB, and HE
caloMetBEFO	reco::CaloMETCollection	MET of all energy deposits in calorimeter towers in EB, EE, HB, HE, HF, and HO
caloMetM	reco::CaloMETCollection	MET of all energy deposits in calorimeter towers in EB, EE, HB, HE, and HF with corrections for muons
MET associations (in RECO and AOD)		
muonMETValueMapProducer, muCorrData	reco::MuonMETCorrectionData	information on how muons were used to correct MET and what associated MIP

		deposits are used
HCAL Noise (RECO and AOD)		
hcalnoise	reco::HcalNoiseRBX	
hcalnoise	HcalNoiseSummary	
Beam Halo (RECO and AOD)		
BeamHaloSummary	reco::BeamHaloSummary	

b-tag intermediate collections (in RECO and AOD)		
btagSoftElectrons	reco::Electron (lxr cvs)	Electron candidates identified by the dedicated btagging SoftElectronProducer, starting from reco::Tracks
softElectronTagInfos	reco::SoftLeptonTagInfo (lxr cvs)	soft electron dedicated TagInfo, containing informations used to b-tag jets due to the presence of a soft electron in the jet
impactParameterTagInfos	reco::TrackIPTagInfo (lxr cvs)	contains information used for btagging about track properties such as impact parameters, decay len, probability to originate from th primary vertex. Uses ak5JetTracksAssociatorAtVertex collection as input.
secondaryVertexTagInfos	reco::SecondaryVertexTagInfo (lxr cvs)	contains the reconstructed displaced secondary vertices in a jet and associated information, uses impactParameterTagInfos as input
ghostTrackVertexTagInfos		
b-tag algorithm result collections (in RECO and AOD)		
softMuonTagInfos	reco::SoftLeptonTagInfo (lxr cvs)	soft muon dedicated TagInfo, containing informations used to b-tag jets due to the presence of a soft muon in the jet
softElectronBJetTags	reco::JetTag (lxr cvs)	results of b-tagging a jet using the SoftElectronTagInfo and the default soft electron tagger, which uses a neural network to combine most electron properties to improve rejection of non-b jets
softMuonBJetTags	reco::JetTag (lxr cvs)	results of b-tagging a jet using the SoftMuonTagInfo and the default soft muon tagger, which uses a neural network to combine most muon properties to improve rejection of non-b jets
jetProbabilityBJetTags	reco::JetTag (lxr cvs)	result of jetProbability algorithm (based on TrackIPTagInfo).
jetBProbabilityBJetTags	reco::JetTag (lxr cvs)	result of jetProbability algorithm in the "jetBProbability" variant.
trackCountingHighPurBJetTags	reco::JetTag (lxr cvs)	Result of track counting algorithm (requiring **three** tracks to have a significance above the

		discriminator). To be used for high purity selection (B eff < 50%, mistag rate < 1%)
trackCountingHighEffBJetTags	reco::JetTag (lxr cvs)	Result of track counting algorithm (requiring **two** tracks to have a significance above the discriminator). To be used for high efficiency selection (B eff > 50%, mistag rate > 1%)
simpleSecondaryVertexHighPurBJetTags	reco::JetTag (lxr cvs)	Uses the flight distance (i.e. distance between a reconstructed secondary vertex and the primary vertex in a jet) as b-tagging discriminator. Secondary vertex is reconstructed with **three** or more tracks.
simpleSecondaryVertexHighEffBJetTags	reco::JetTag (lxr cvs)	Uses the flight distance (i.e. distance between a reconstructed secondary vertex and the primary vertex in a jet) as b-tagging discriminator. Secondary vertex is reconstructed with **two** or more tracks. Can be configured to return the value or significance in 2d and 3d, optionally corrected for the boost at the SV - works up to a maximum secondary vertex finding efficiency of ~70% in b-jets
combinedSecondaryVertexBJetTags	reco::JetTag (lxr cvs)	Result of application of a likelihood estimator to the tagging variables for the three possible algorithm outcomes (tracks only, pseudo vertex from at least two tracks or successful secondary vertex fit), obtained from impactParameterTagInfos and secondaryVertexTagInfos
combinedSecondaryVertexMVABJetTags	reco::JetTag (lxr cvs)	uses the PhysicsTools/MVAComputer framework to compute a discriminator from the impactParameterTagInfos and secondaryVertexTagInfos with an uptodate calibration from the the CMS conditions database, using a neural network instead of a likelihood ratio in case an actual secondary vertex was reconstructed
ghostTrackBJetTags		

tau-tag collections (in RECO and AOD)

missing

Vertex collections (in RECO and AOD)		
offlinePrimaryVertices	reco::VertexCollection (lxr cvs)	Primary vertex reconstructed using the tracks taken from the generalTracks collection
offlinePrimaryVerticesWithBS	reco::VertexCollection (lxr cvs)	Primary vertex reconstructed using the tracks taken from the generalTracks collection, and imposing the offline beam spot as a constraint in the fit of the vertex position.
VertexCompositeCandidate collections (in RECO and AOD)		
generalV0Candidates:Kshort	reco::VertexCompositeCandidateCollection (lxr cvs)	K0S candidate collection reconstructed using the tracks taken from the generalTracks collection
generalV0Candidates:Lambda	reco::VertexCompositeCandidateCollection (lxr cvs)	Lambda0 candidate collection reconstructed using the tracks taken from the generalTracks collection

Pixel vertex collections (in RECO and AOD)		
pixelTracks	reco::TrackCollection	(proto)tracks created from two or three hits in Pixel detector
pixelVertices	reco::VertexCollection	primary vertices reconstructed from pixel tracks
Pixel vertex collections (in RECO only)		
pixelTracks	reco::TrackExtraCollection	

HLT trigger collections (in RECO and AOD)		
TriggerResults	edm::TriggerResults	One EDproduct per HLT table: the HLT decision - global, and for each HLT trigger path: Ready (ie, Not Run), Pass (Accept), Fail (Reject), Error (Exception)
hltTriggerSummaryAOD	trigger::TriggerEvent	One EDproduct per HLT table - contains L3 collections and L3 filter decisions
hltTriggerSummaryRAW	trigger::TriggerEventWithRefs	One EDproduct for each HLT path in the HLT table - contains Ref links pointing to the original HLT collections and records filter decisions
HLT filter module label	trigger::TriggerFilterObjectWithRefs	one of these by every HLT filter module - transient - used to provide information put into persistent TriggerEvent or TriggerEventWithRefs
hltTimer	edm::EventTime containing a vector of edm::ModuleTime	Simple class for Rate/CPU/Wall time benchmarking of algorithms - generic, not HLT specific
hltPathTimer	HLTPerformanceInfo	Advanced class for Rate/CPU/Wall time benchmarking of the HLT table

Particle flow collections (in RECO and AOD)

particleFlow	reco::PFCandidateCollection	Particle Flow Candidates (refers to reco::PFBlock's)
particleFlow:electrons	reco::PFCandidateCollection	Particle Flow Electron Candidates without any selection (refers to reco::PFBlock's)
pfElectronTranslator:pf	reco::SuperClusterCollection	Particle Flow Super-clusters of PF-electrons
pfElectronTranslator:pf	reco::CaloClusterCollection	Basic clusters of PF electron super-clusters
pfElectronTranslator:pf	reco::PreshowerClusterCollection	Preshower clusters of PF electron super-clusters
pfPhotonTranslator:pfphot	reco::SuperClusterCollection	Particle Flow Super-clusters of photons imported in PF
pfPhotonTranslator:pfphot	reco::CaloClusterCollection	Basic clusters of PF photon super-clusters
pfPhotonTranslator:pfphot	reco::PreshowerClusterCollection	Preshower clusters of PF photon super-clusters
pfPhotonTranslator:pfphot	reco::PhotonCollection	photons imported in PF translated into the RECO format
particleFlow:electrons		ValueMap(GsfElectronRef,PFCandidatePtr)
particleFlow:photons		ValueMap(PhotonRef,PFCandidatePtr)

Particle flow collections (in RECO only)		
particleFlowCluster(ECAL)	reco::PFClusterCollection	ECAL clusters
particleFlowCluster(HCAL)	reco::PFClusterCollection	HCAL clusters
particleFlowCluster(PS)	reco::PFClusterCollection	preshower clusters
particleFlowBlock	reco::PFBlockCollection	Particle Flow Blocks (refers to reco::Track's and reco::PFCluster's)
trackerDrivenElectronSeeds:preid	reco::PreIdCollection	Information on the tracker-driven electron seed
pfElectronTranslator:pf	reco::GsfElectronCollection	PF electron reconstructed translated in the GsfElectron format

Questions:

- As reported by Colin, the links to the class references have some strange characters like dd/d5b and numbers in them. July 2008: This has now been solved see WorkBookContributors#LinkRefMan
- I wanted to add a link to the default definition of the module, but got lost, i.e. where to point in cvs for ctfWithMaterialTracks? July 2008: this is being investigated, doxygen may be able to do this with python config files.

Review Status

CMSUserSupport - 24 Apr 2007: Page author

Responsible: CMSUserSupport

This topic: CMSPublic > SWGuideDataFormatTable
Topic revision: r59 - 2012-11-26 - TaiSakuma