
DaVinci status

Juan Palacios (Nikhef)

LHCb software week

15-19 June 2009

- Latest releases
- Particle->MC association
- PV re-fitting and Particle->PV association
- New stripping framework
- New HLT stuff
- Improvements to MicroDST
- Developments in DecayTreeTuple

- Latest releases
- Particle->MC association Dedicated talk today
- PV re-fitting and Particle->PV association
- New stripping framework Dedicated talk tomorrow
- New HLT stuff
- Improvements to MicroDST Dedicated talks
this morning
- Developments in DecayTreeTuple

- Since last SW week,
 - v23 series, latest version v23r1p1
 - v23r0 Mainly driven by changes to HLT
 - FEST: This required some urgency
 - No time to test many offline additions
 - v23r1 first STEP09 version
 - Driven by new stripping framework
 - All PhysSel packages removed
 - Most v23r0 offline additions tested

- New tools to replace DaVinciAssociators
 - Developed in collaboration with Vava
- Interfaces designed for ease of use
 - Non-expert “give me one MCParticle and I don’t care how”
 - Expert “give me some more info so I can tune the association for the non-experts
 - Expert “let me plug in my new MC association logic”

- More user control
 - Tools “stupid”
 - They should do what it says on the label and no more
 - Users grab tools, pass them some input, and can do whatever they want with the output
 - Framework approach
 - Extensions can be achieved with very few lines of code
- More details/examples in dedicated talk later today

- PV re-fitting problem is an old one
 - Tools for re-fitting PV after removing tracks from particle have existed for some time
 - Yuehong, Maruisz
 - But how to incorporate that into particle-making and selecting code?
 - Many particle, many PV events?
 - Best PV after re-fitting might not be best PV re-fitted!
 - How to marry re-fitting with “best PV” mechanism of DVAlgorithm?
-

- See new Wiki
 - <https://twiki.cern.ch/twiki/bin/view/LHCb/Particle2PV>
 - Basic principle:
 - Ask for a best PV and get P->PV weighted relations that are stored locally
 - The relations are put on TES so following DVAlgorithms can use them
 - The DVAlgorithms can take tables as input

- Would want to be able to make the “best PV” decision based on re-fitted PVs
 - More than one re-fitted PV per particle
- Re-fitted PVs only mean something in relation to a particle
 - Need to keep them linked
- Also want to be able to re-fit PVs for selections when no best PV has been asked for
 - No cuts involving PV
- Two approaches
 - Automatic in DVAlgorithm
 - Stand-alone GaudiAlgorithms

- DVAlgorithm can re-fit all PVs each time a best PV is asked for
 - All PVs re-fitted
 - Best one found with IRelatedPVFinder
 - Relation between particle and best re-fitted PV stored
 - For each selected particle, the best re-fitted PV and relation is saved to the TES
 - Just set **MyAlgo.ReFitPVs = True**

- The PVReFitterAlg is a simple GaudiAlgorithm
 - Take TES locations of input Particles and PVs
 - Refit all vertices for all particles
 - Put re-fitted vertices and Particle->Refitted PV relations on user-defined TES locations
 - Relations given artificial weight to maintain same order of PVs as for original PV container
 - No concept of “best PV” at this stage

- The PVRelatorAlg is another simple GaudiAlgorithm
 - Take TES locations of input Particles
 - Take TES location of PVs or P->PV relations (our case)
 - Use an IRelatedPVFinder to get re-ordered relations
 - Put relations in user-defined TES location
 - *Apologies for the lack of doxygen documentation*

- Use PVReFitterAlg – PVRelatorAlg tandem:
 - Example taken from `MicroDSTExample/options/TestMicroDSTMake.py`

```

From Configurables import PVReFitterAlg, PVRelatorAlg
mainLocation = "Phys/SomeSelectionOutputPath"
# re-fit PVs from standard location
PVReFitter = PVReFitterAlg("PVReFitterAlg")
PVReFitter.ParticleInputLocation = mainLocation+"/Particles"
PVReFitter.VertexOutputLocation = mainLocation+"/RefittedVertices"
p2ReFitPVRelationsLoc = mainLocation+"/Particle2ReFittedVertexRelations"
PVReFitter.P2VRelationsOutputLocation = p2ReFitPVRelationsLoc
# re-sort the relations using an IRelatedPVFinder
PVRelator = PVRelatorAlg()
PVRelator.P2PVRelationsInputLocation = p2ReFitPVRelationsLoc
p2pvSortedRelationsLoc = mainLocation + "/P2ReFitPVSortedRelations"
PVRelator.P2PVRelationsOutputLocation = p2pvSortedRelationsLoc
MySequence.Members += [PVReFitter, PVRelator]

```

- Both methods result in the production of a weighted Particle -> VertexBase relations table plus some re-fitted PVs on the TES
- **NEVER** access the PVs directly!
 - Slightly different versions of the same PV
- The PVs mean nothing unless related to a particle
- **ALWAYS** get them via the relations!
 - Either pass on to a DVAlgorithm or use directly

- PhysSel is no more
 - Well, it is, but it isn't in the release
 - Need to getpack packages if want to run an old pre-selection
 - \$DAVINCIROOT/options/DaVinci.py broken
 - Fixed in CVS
- New stripping framework (A. Poluektov)
 - Based on Hlt2Lines concept and code
 - Visit the wiki:
<https://twiki.cern.ch/twiki/bin/view/LHCb/StrippingLines>
 - See Anton's talk tomorrow!

- All MicroDSTAlgorithms now generic
 - Last case is cloning of objects linked via relations tables
 - Particle->MCParticle, Particle->Primary vertex
- Added HltDecReports
 - Some issues with reflex dictionaries solved in DV v23r1p1
- New PhysMicroDST() configurable
 - Makes creating a MicroDST without understanding what you're doing much easier!
- Wiki updated
 - <https://twiki.cern.ch/twiki/bin/view/LHCb/MicroDST>
- See my talk from this morning

- MicroDST
 - <https://twiki.cern.ch/twiki/bin/view/LHCb/MicroDST>
- Particle -> MC association
 - <https://twiki.cern.ch/twiki/bin/view/LHCb/Particle2MC>
- Particle -> PV relating
 - <https://twiki.cern.ch/twiki/bin/view/LHCb/Particle2PV>
- Stripping framework
 - <https://twiki.cern.ch/twiki/bin/view/LHCb/StrippingLines>