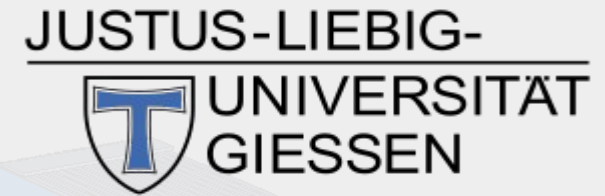


# XLIX PANDA Collaboration Meeting (GSI)



## Event Filter for PandaRoot / FairRoot

**This talk is an update of my talk at the PANDA CM in March 2014**

**Martin J. Galuska, Katja Kleeberg, J. Sören Lange, Wolfgang Kühn  
Justus Liebig Universität Gießen**

This work was supported in part by BMBF (06GI9107I), HGS-HIRe for FAIR and the LOEWE-Zentrum HICforFAIR.



Helmholtz International Center



Bundesministerium  
für Bildung  
und Forschung

# Outline

- Motivation
- Concept and Criteria
  - Predefined filters
  - User-defined filters
- Classes
- How to
  - Use predefined filters
  - (Implement user-defined filters)
- Summary and Outlook



# Motivation

# Motivation for an Event Filter

- **Problem:** Imagine you want to...
  - ... run **PandaRoot full or fast simulation** only for **specific** subsets of **events from some event generator(s)**.
  - ... **save time and disk space**.
- **Solution: Event filter**
  - **Look at the particles produced by the event generator(s) BEFORE** they are **transported** through the detector model and **BEFORE** digitization, tracking, PID, ...
  - **If** a generated **event is interesting**, **run simulation** and analysis on this event. **If not, discard the event** and **rerun the event generator(s)**.
- **Example:**
  - Extraction of signal events from general generator.
  - Background studies (with dpm) for specific signal channel.
  - Subdetector occupancy/performance studies.

# Event Filter

## Concept and Criteria

# Event Filter Concept in PandaRoot

- **Event filter framework for full and fast simulation**
  - All event generators (+combinations) supported.
  - Connect multiple filters with **AND, OR, NOT**.
  - Veto filters.
  - Count number of **generated / accepted / failed events**.
- **Flexible predefined event filters**
  - Filter on single particle properties.
  - Filter on invariant masses of  $\leq 5$  particles.
- **Framework for user-defined event filters**
  - Enables users to implement their own specific event filter(s).
  - Only *Bool\_t EventMatches(Int\_t evtNr)*; needs to be implemented.
  - RHO package can be used.

# Event Filter Framework – Status March 2014

Already implemented  
Ongoing  
Planned

- Event filtering capabilities.
  - And / or / not for connecting multiple filters.
- Virtual filter class for FairRoot.
- General use predefined filter for single particle properties.
- Framework for user-defined event filters (with tools from analysis).
- Debugging and QA.

# Event Filter Framework – Status June 2014

Already implemented  
Ongoing  
Planned

- Event filtering capabilities.
  - And / or / not for connecting multiple filters.
- Virtual filter class for FairRoot.
- General use predefined filter for single particle properties.
- Framework for user-defined event filters (with tools from analysis).
- Debugging and QA.





# Criteria in Predefined Event Filters – March 2014

Already implemented  
Ongoing / Planned  
Won't be Implemented

- Single particle properties
  - Momentum (total / transversal / z)
  - Angles (theta, phi)
  - Vertex (z / rho / r)
  - (List of) PDG code(s) / Charge (neutral / charged / + / –)
  - Angles + momenta in center of mass system for arbitrary 4 vectors
- Filter on particle combinations
  - Invariant masses (of arbitrary combinations)
  - + same criteria as for single particles

# Criteria in Predefined Event Filters – June 2014

Already implemented  
Ongoing / Planned  
Won't be Implemented

- Single particle properties
  - Momentum (total / transversal / z)
  - Angles (theta, phi)
  - Vertex (z / rho / r)
  - (List of) PDG code(s) / Charge (neutral / charged / + / –)
  - **Angles + momenta in center of mass system for arbitrary 4 vectors**
- Filter on particle combinations
  - Invariant masses (of arbitrary combinations)
  - **+ same criteria as for single particles**



# Event Filter Classes

Filter framework  
Base class for  
user-defined filters  
Predefined filter

- **...FilteredPrimaryGenerator** (FairPrimaryGenerator)
  - Adds event filtering capabilities to FairPrimaryGenerator
- **...EvtFilterParams** (TObject)
  - Helper class for saving filter statistics
- **...EvtFilter** (TNamed)
  - Virtual base class for event filters on FairRoot level (based on TParticle)
- **...EvtFilterOnSingleParticleCounts** (FairEvtFilter)
  - Predefined event filter for single particle properties

More details in  
backup slides

Filter framework  
Base class for  
user-defined filters  
Predefined filter

- **...EvtFilter** (**FairEvtFilter**)
  - Adds PandaRoot specific functionality to FairEvtFilter.
  - Virtual base class for event filters on PandaRoot level (based on **RhoCandidate**).
- **...EvtFilterOnInvMassCounts** (**PndEvtFilter**)
  - Predefined event filter for invariant masses of  $\leq 5$  particles

More details in  
backup slides

# How to use the event filters

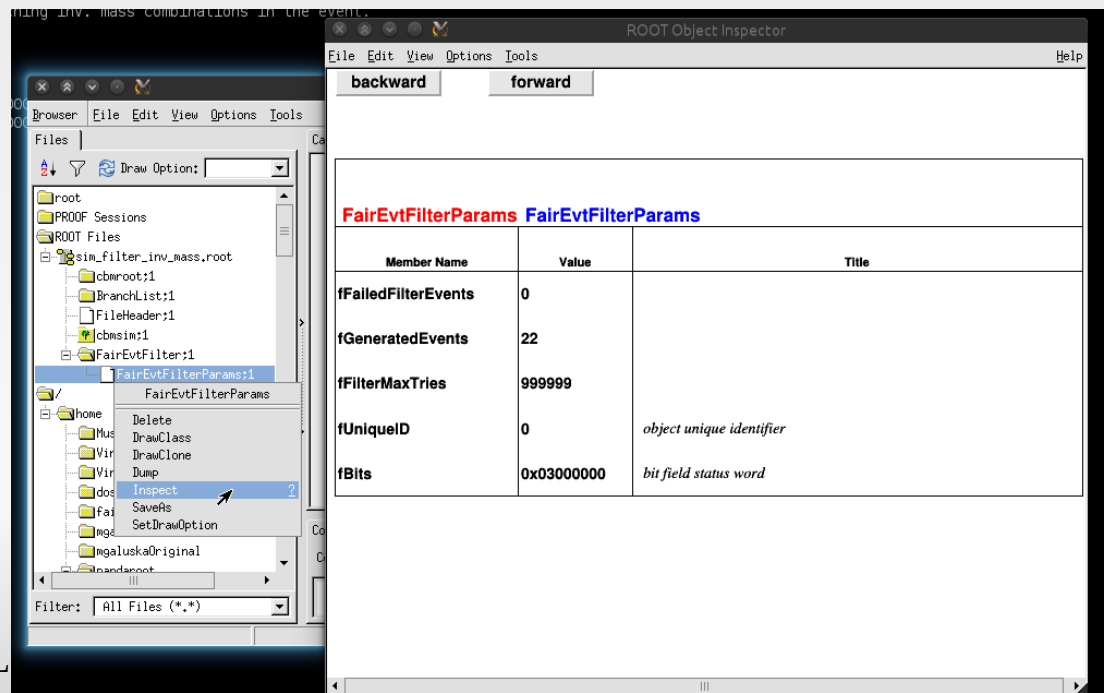
# First: A Word of Caution

- The event filter(s) run(s) BEFORE particle transport!
  - No material effects
  - No tracking efficiencies / momentum resolutions
  - No PID inefficiencies
- **Only skip events if you are absolutely sure you don't need them!**
- Recipy for study of general DPM background:
  - Run simulation with ALL events from DPM.
  - Apply event selection in analysis as for signal events.
  - Look at primaries from surviving events to find event topologies.
  - Now you can save time and disk space by using the event filter to produce more background events.

# Necessary Changes in sim Macro

- Use **FairFilteredPrimaryGenerator** instead of FairPrimaryGenerator
  - FairFilteredPrimaryGenerator\*** primGen = new **FairFilteredPrimaryGenerator()**;  
fRun->SetGenerator(primGen);
- Add one line after fRun->Run(nEvents);  
to write the statistics of the event filtering into the output root file.
  - //----- Run the Simulation -----  
fRun->Run(nEvents);  
**primGen->WriteEvtFilterStatsToRootFile();**

- I suggest to make this the standard in macro/run.



The screenshot shows the ROOT Object Inspector window. The left pane displays a tree view of the ROOT file structure, with 'FairEvtFilterParams:1' selected. The right pane shows the object's details, including a table of member variables and their values.

Member Name	Value	Title
fFailedFilterEvents	0	
fGeneratedEvents	22	
fFilterMaxTries	999999	
fUniqueID	0	object unique identifier
fBits	0x03000000	bit field status word



# sim Macro Structure for Background Production

```
// ...
```

```
FairFilteredPrimaryGenerator* primGen = new FairFilteredPrimaryGenerator();  
fRun->SetGenerator(primGen);
```

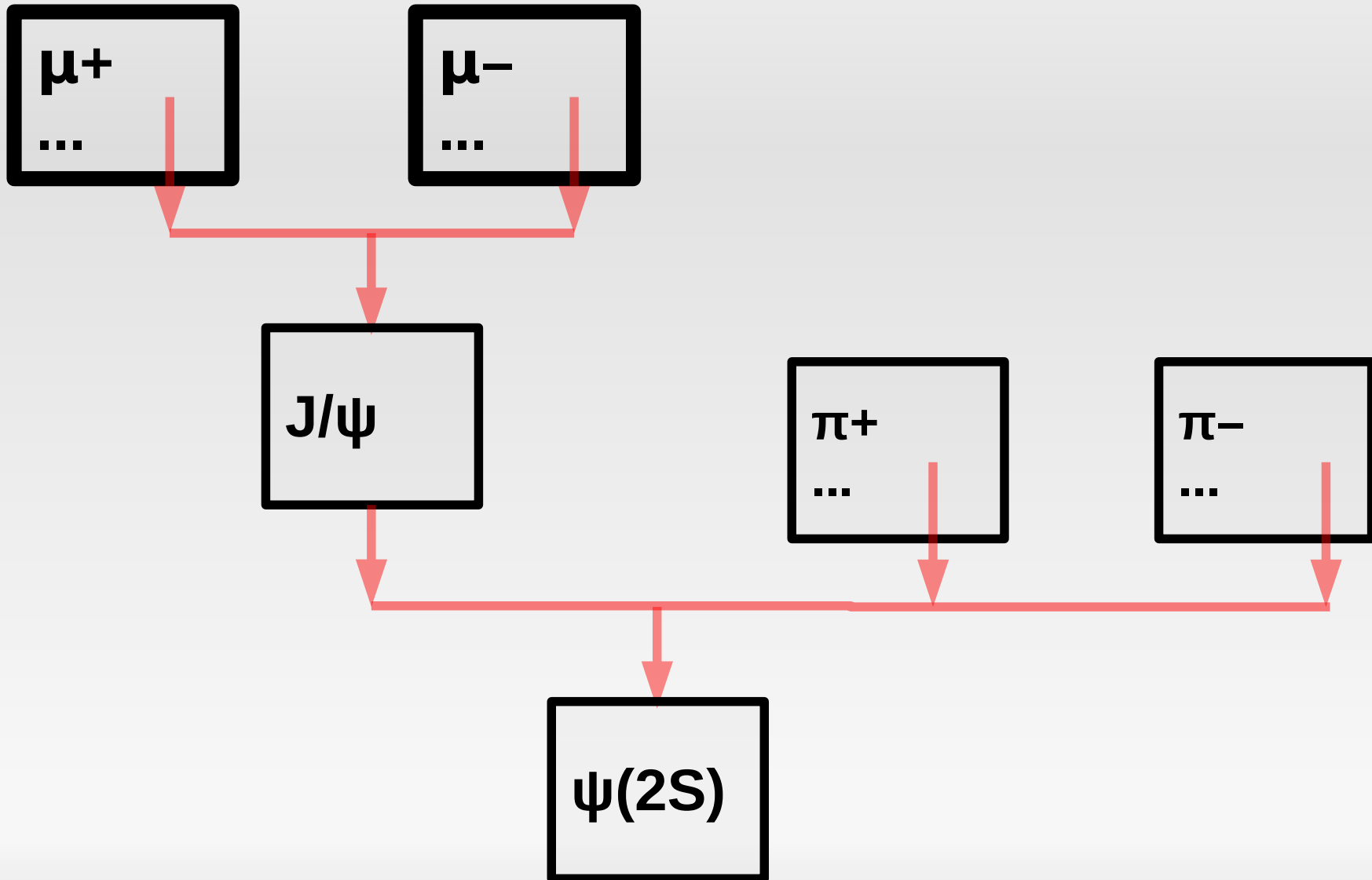
```
// insert the desired event generator(s), let's say DpmDirect  
PndDpmDirect *Dpm= new PndDpmDirect(mom,1);  
primGen->AddGenerator(Dpm);
```

```
// optionally insert some event filters (if not, no filtering is done)
```

```
// ...
```

```
//----- Run the Simulation -----  
fRun->Run(nEvents);  
primGen->WriteEvtFilterStatsToRootFile();
```

# Example Channel: $\psi(2S) \rightarrow J/\psi (\mu^+ \mu^-) \pi^+ \pi^-$



Example:  $\psi(2S) \rightarrow J/\psi (\mu^+ \mu^-) \pi^+ \pi^-$   
Require min. 2 charged particles

```
FairEvtFilterOnSingleParticleCounts* myFilter = new  
FairEvtFilterOnSingleParticleCounts();
```

```
myFilter->AndMinCharge( 2, FairEvtFilter::kCharged );
```

```
primGen->AndFilter(myFilter);
```

Example:  $\psi(2S) \rightarrow J/\psi (\mu^+ \mu^-) \pi^+ \pi^-$   
Require min. 2, max. 6 charged particles

```
FairEvtFilterOnSingleParticleCounts* myFilter = new  
FairEvtFilterOnSingleParticleCounts();
```

```
myFilter->AndMinMaxCharge( 2, 6, FairEvtFilter::kCharged );
```

```
primGen->AndFilter(myFilter);
```

Example:  $\psi(2S) \rightarrow J/\psi (\mu^+ \mu^-) \pi^+ \pi^-$

Require **min. 1+**, **min. 1-**, max. 6 charged particles

```
FairEvtFilterOnSingleParticleCounts* myFilter = new  
FairEvtFilterOnSingleParticleCounts();
```

```
myFilter->AndMinCharge( 1, FairEvtFilter::kPlus );
```

```
myFilter->AndMinCharge( 1, FairEvtFilter::kMinus );
```

```
myFilter->AndMaxCharge( 6, FairEvtFilter::kCharged );
```

```
primGen->AndFilter(myFilter);
```

Example:  $\psi(2S) \rightarrow J/\psi (\mu^+ \mu^-) \pi^+ \pi^-$

Require **min. 1  $\mu^+$  /  $\pi^+$  /  $K^+$** , **min. 1  $\mu^-$  /  $\pi^-$  /  $K^-$** ,  
**max. 6 charged particles**

```
FairEvtFilterOnSingleParticleCounts* myFilter = new  
FairEvtFilterOnSingleParticleCounts();
```

```
myFilter->AndMinPdgCodes( 1, -13, 211, 321 );
```

```
myFilter->AndMinPdgCodes( 1, 13, -211, -321 );
```

```
myFilter->AndMaxCharge( 6, FairEvtFilter::kCharged );
```

```
primGen->AndFilter(myFilter);
```

pdg codes as used by EvtGen  
(trunk/pgenerators/EvtGen/EvtGen/Private/evt.pdl)

Example:  $\psi(2S) \rightarrow J/\psi (\mu^+ \mu^-) \pi^+ \pi^-$

Require **min. 1  $\mu^+$  /  $\pi^+$  /  $K^+$** , **min. 1  $\mu^-$  /  $\pi^-$  /  $K^-$** , **min. 1 soft  $\pi^+$  /  $K^+$** , **min. 1 soft  $\pi^-$  /  $K^-$** , **max. 6 charged particles**

```
FairEvtFilterOnSingleParticleCounts* maxCharged = new  
FairEvtFilterOnSingleParticleCounts();  
maxCharged->AndMaxCharge( 6, FairEvtFilter::kCharged );
```

```
FairEvtFilterOnSingleParticleCounts* normalP = new  
FairEvtFilterOnSingleParticleCounts();  
normalP->AndPRange( 0.0, 1.0);  
normalP->AndMinPdgCodes( 1, -13, 211, 321 );  
normalP->AndMinPdgCodes( 1, 13, -211, -321 );
```

```
FairEvtFilterOnSingleParticleCounts* softP = new FairEvtFilterOnSingleParticleCounts();  
softP->AndPRange( 0.8, 100.0);  
softP->AndMinPdgCodes( 1, 211, 321 );  
softP->AndMinPdgCodes( 1, -211, -321 );
```

```
primGen->AndFilter(maxCharged);  
primGen->AndFilter(normalP);  
primGen->AndFilter(softP);
```

Logically combine multiple event filters with AND, OR, ANDNOT, ORNOT  
Order of filter addition to primGen is important!

Example:  $\psi(2S) \rightarrow J/\psi (\mu^+ \mu^-) \pi^+ \pi^-$

Require min. 1  $\mu^+$  /  $\pi^+$  /  $K^+$ , min. 1  $\mu^-$  /  $\pi^-$  /  $K^-$ , min. 1 soft  $\pi^+$  /  $K^+$ , min. 1 soft  $\pi^-$  /  $K^-$ , not more than 6 charged particles

```
FairEvtFilterOnSingleParticleCounts* min7Charged = new  
FairEvtFilterOnSingleParticleCounts();  
min7Charged->AndMinCharge( 7, FairEvtFilter::kCharged );
```

```
FairEvtFilterOnSingleParticleCounts* normalP = new  
FairEvtFilterOnSingleParticleCounts();  
normalP->AndPRange( 0.0, 1.0);  
normalP->AndMinPdgCodes( 1, -13, 211, 321 );  
normalP->AndMinPdgCodes( 1, 13, -211, -321 );
```

```
FairEvtFilterOnSingleParticleCounts* softP = new FairEvtFilterOnSingleParticleCounts();  
softP->AndPRange( 0.8, 100.0);  
softP->AndMinPdgCodes( 1, 211, 321 );  
softP->AndMinPdgCodes( 1, -211, -321 );
```

```
primGen->AndNotFilter(min7Charged);  
primGen->AndFilter(normalP);  
primGen->AndFilter(softP);
```



Example:  $\psi(2S) \rightarrow J/\psi (\mu^+ \mu^-) \pi^+ \pi^-$

Require min. 1  $\mu^+ / \pi^+ / K^+$ , min. 1  $\mu^- / \pi^- / K^-$ , min. 1 soft  $\pi^+ / K^+$ , min. 1 soft  $\pi^- / K^-$ , veto for more than 6 charged particles

```
FairEvtFilterOnSingleParticleCounts* min7Charged = new  
FairEvtFilterOnSingleParticleCounts();  
min7Charged->AndMinCharge( 7, FairEvtFilter::kCharged );
```

```
FairEvtFilterOnSingleParticleCounts* normalP = new  
FairEvtFilterOnSingleParticleCounts();  
normalP->AndPRange( 0.0, 1.0);  
normalP->AndMinPdgCodes( 1, -13, 211, 321 );  
normalP->AndMinPdgCodes( 1, 13, -211, -321 );
```

```
FairEvtFilterOnSingleParticleCounts* softP = new FairEvtFilterOnSingleParticleCounts();  
softP->AndPRange( 0.8, 100.0);  
softP->AndMinPdgCodes( 1, 211, 321 );  
softP->AndMinPdgCodes( 1, -211, -321 );
```

```
primGen->AddVetoFilter(min7Charged);  
primGen->AndFilter(normalP);  
primGen->AndFilter(softP);
```

Veto filters have higher priority than regular event filters.  
If event matches min. 1 veto filter, it is skipped immediately.

Example:  $\psi(2S) \rightarrow J/\psi (\mu^+ \mu^-) \pi^+ \pi^-$

Require min. 1  $\mu^+ \mu^-$  cand. Pair with inv. Mass in  $J/\psi$  region

```
// filter on mu+ mu- invariant mass
```

```
PndEvtFilterOnInvMassCounts* mmlnv= new PndEvtFilterOnInvMassCounts();
```

```
mmlnv->SetPdgCodesToCombine( -13, 13);
```

```
mmlnv->SetMinMaxInvMass( 2.0, 4.0 );
```

```
mmlnv->SetMinCounts(1);
```

```
// same filters as on last slide
```

```
primGen->AndFilter(mmlnv);
```

Combines all + with all - charged particles in the event using  $\mu^+ / \mu^-$  mass hypothesis.

Events with min. 1 combination with inv. mass. between 2.0 and 4.0 GeV match.

# Tutorial



- Tutorial on event filter usage and implementation of user-defined event filters
  - <https://panda-wiki.gsi.de/foswiki/bin/view/Computing/PandaRootEventFilterTutorial>
  - **Currently still under construction**



# Summary and Outlook

# Summary and Outlook

- Event filtering was successfully implemented in PandaRoot.
  - Code in trunk/pgenerators/eventFilter.
  - Compiled by default.
  - **Please use the event filter and report bugs.**
- Tutorial on usage of predefined event filters and implementation of user-defined event filters.
  - <https://panda-wiki.gsi.de/foswiki/bin/view/Computing/PandaRootEventFilterTutorial>
- I will provide bug fixes.
- Inclusion on FairRoot level possible.
  - Exception: Predefined event filter on invariant masses.

# Thank You!

- We hope the features will be useful for background analysis, detector performance studies and many more applications in the future and can be adopted on the FairRoot level into the `trunk/base/sim` code.



# Backup Slides

# Event Filter Criteria

- On **arbitrary particle combinations**
  - **Invariant masses**
- On **individual particles or arbitrary particle combinations**
  - **Momentum** (total / transversal / z) [in lab system]
  - **Angles** (theta, phi) [in lab system]
  - **Vertex** (z / rho / r)
  - **PDG code / Charge** (neutral / + / - / charged )
  - **Angles + momenta in arbitrary center of mass system**



- **...FilteredPrimaryGenerator** (FairPrimaryGenerator)
  - Adds event filtering capabilities to FairPrimaryGenerator
  - Manages event filters and veto filters
  - Gets TParticle from generators and feeds them into event filters
  - Calls event filters, logically connects their output
- **...EvtFilterParams** (TObject)
  - Helper class for saving filter statistics
- **...EvtFilter** (TNamed)
  - Virtual base class for event filters on FairRoot level (based on TParticle)
- **...EvtFilterOnSingleParticleCounts** (FairEvtFilter)
  - Predefined event filter for single particle properties
  - Multiplicity of charge states or pdg code lists
  - Optionally define angles, momenta, vertices

Filter framework  
Base class for  
user-defined filters  
Predefined filter

- **...EvtFilter** (**FairEvtFilter**)
  - Adds PandaRoot specific functionality to FairEvtFilter.
    - *Bool\_t FillList ( RhoCandList& rhoOutList, Int\_t inPdgCode, Double\_t pdgCodeCharge = kNoChargeSpecified );*
    - returns a list of particles
    - which have the same charge as charge(inPdgCode)
    - mass hypothesis is set to m(inPdgCode)
  - Virtual base class for event filters on PandaRoot level (based on **RhoCandidate**).
- **...EvtFilterOnInvMassCounts** (**PndEvtFilter**)
  - Predefined event filter for invariant masses of  $\leq 5$  particles
  - User input
    - pdg codes to be combined
    - Invariant mass region
    - Desired multiplicity range

Filter framework  
Base class for  
user-defined filters  
Predefined filter

# On the Event Filter Usage: **A Word of Caution**

- The event filter(s) **only looks at particles** produced by the **event generator(s)**.
  - It does not know about any non-generator created particles (because it analyses the event **BEFORE** the transport engine is run)!
  - It does not take material effects / interaction with detector / tracking efficiencies / momentum resolutions / PID / etc. into account.
- Be careful **not to filter too tightly!** Be careful when you **extract physical meaning** from an analysis of filtered events!
- In case of the DPM generator, the user should analyse ALL events from DPM with sufficient statistics and based on the results, (s)he can decide that mainly certain event topologies contribute (after cuts) to the background events for the analysed channel.
- Once such specific events were identified, the filter can be useful in saving simulation time and computer resources.