Signal-background mixing in FairRoot

M. Al-Turany

Remarks

- Simulation is not effected by the new features
- Event time can be set during or after the simulation
- Mixing can be done on event number or event time
- One background Chain can be used
- No limit on the number of signals that can be used

Event Time In Simulation

Can be set via the FairPrimaryGenerator :

- Set the min and max limit for event time in ns
  SetEventTimeInterval(Double_t min, Double_t max)
- Set the mean time for the event in ns
  SetEventMeanTime(Double_t mean)
- Set the time function for event
  SetEventTime(TF1 timeProb)*

Event Time after simulation

Can be set via the FairRunAna

- Set the min and max limit for event time in ns

  SetEventTimeInterval(Double_t min, Double_t max)

  (Time set via Uniform Random between min and max)

- Set the mean time for the event in ns

  SetEventMeanTime(Double_t mean)

  (Event time is an exponential deviate.)

Event Time

- In simulation event time is saved in
  - *FairMCEventHeader*
- After simulation it is in
  - *FairEventHeader*
- In all cases event time can be access via:
  - *FairRootManager::GetEventTime()*

File Header

- Run Id
- List of TObjStrings presenting the class names of tasks used to produce this file
- List of FairFileInfo presenting the input files used to produce this file

FairFileInfo class

- Full path of the file
- Size of file in bytes
- File identifier used
- File order in the chain

Event Header

- Run Id
- Event Time
- Input file identifier, the file description is in the File header
- Monte-Carlo entry number from input chain

Signal-Background mixing

- After simulation
- At the moment we only check that the all input trees contain the same branches (further checks needed!)
- All can be done from the macro

Example of a mixing digi macro

See "Pandaroot/macro/run/example_mix"

- Setting the input files:

```
//** Set BG file */
fRun->SetBackgroundFile("sim_stt_bg.root");

//** Set first signal file */
 fRun->SetSignalFile("sim_stt_s1.root",1);

//** Set second signal file */
fRun->SetSignalFile("sim_stt_s2.root",2);
```

Example of a mixing digi macro

- Adding more files to the signal and background chains:

```
//** Set BG file */
fRun->AddBackgroundFile("sim_stt_bg1.root");

//** Set first signal file */
 fRun->AddSignalFile("sim_stt_s1_1.root",1);

//** Set second signal file */
fRun->AddSignalFile("sim_stt_s2_1.root",2);
```

Example:
Mix using entries

- For each ~20 entries background one entry from signal chain 1 should be read

  fRun->BGWindowWidthNo(20,1)

- for each ~30 entries background one entry from signal chain 2 should be read

  fRun->BGWindowWidthNo(30,2)

Example: Mix using time

- Set the event mean time, event time will be a random number generated from (1/T)exp(-x/T)

  fRun->SetEventMeanTime(10);

- Each ~100 ns background 1 entry from signal chain 1 will be read

  fRun->BGWindowWidthTime(100,1);

- Each ~60 ns background 1 entry from signal chain 2 will be read

  fRun->BGWindowWidthTime(60,2);

Running the macro

[INFO] Maximum No of Event was set manually to : 120 , we will check if there is enough entries for this!!

[INFO] Signal chain No 1 has : 2 entries

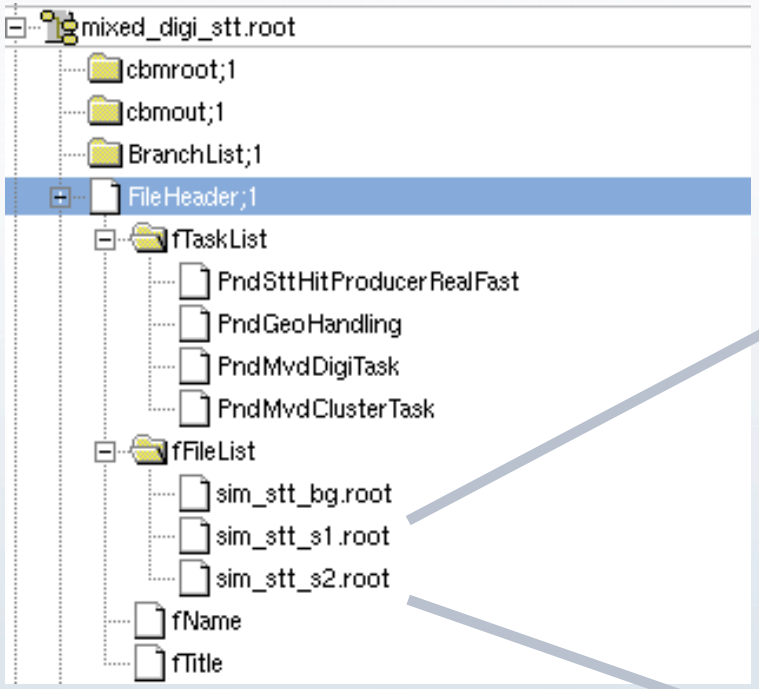
[WARNING] No of Event in signal chain 1 is not enough, the maximum event number will be reduced to : 20

[INFO] Signal chain No 2 has : 20 entries

[WARNING] No of Event in Background chain is not enough for all signals in chain 2

[INFO] Maximum No of Event will be set to : 20

Output




```
class=FairFileInfo
fPath          /pandaroot/macro/run/example_mix
fSize          52607
fIdentifier    1
fInChainId     0
fName          sim_stt_s1.root    object identifier
```

```
class=FairFileInfo
fPath          /pandaroot/macro/run/example_mix
fSize          243274
fIdentifier    2
fInChainId     0
fName          sim_stt_s2.root    object identifier
```

To Do

- More checks before mixing:
  - Parameters
  - Geometry
- Mixing signal in a sub-set of the detector with the full back ground simulation