

---

Subject: Multiple input files / trees

Posted by [Volker Frieze](#) on Thu, 22 Mar 2012 09:59:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

I would like to ask about the status of a topic that was discussed some time ago, concerning the possibility to have two (or more) input files of the same data level (the same branches) for a reconstruction run.

The concrete example is in our case to use two files with MC (transport) data and merge them before digitisation. A typical use case for us is to overlay background events (e.g. from UrQMD with some thousand tracks) with signal events (e.g. a J/psi decay with only two daughter tracks). Transporting UrQMD is very CPU and time expensive, while the transport of signals is not. So, one would like to re-use the same transported UrQMD events as background for different signals.

This demand can be generalised to using multiple files with the same branches at any stage of the reconstruction.

I recall that some years ago, Denis made some attempts to solve this, but I think there were major difficulties, and the issue was dropped eventually.

Are there plans to take this issue up again on the framework level? Or is the feature already solved without my being aware?

Best regards,

Volker

---

Subject: Re: Multiple input files / trees

Posted by [Denis Bertini](#) on Thu, 22 Mar 2012 10:38:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi Volker

The way the merging was done several years ago is based on a virtual function of the FairDetector that has to be implemented by the concrete detector in order to proceed for the merging.

This function was working when we implemented it ( if i remenber something like 2004 or 2005 ) but never used really. Instead people i guess were using the generator and transporting Urqmd and signal all together.

Since it was not used, this way of merging was not maintained and supported by us.

Now according to Mohammad , there are other way to do this merging not directly on Geant data but on the level of Digis which includes time based simulation. Differentiation between background and signal are then handled by the FairLink

classes.

Regards

---

---

Subject: Re: Multiple input files / trees

Posted by [Volker Frieze](#) on Thu, 22 Mar 2012 10:59:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Dear Denis,

Quote: This function was working when we implemented it ( if i remember something like 2004 or 2005 ) but never used really. Instead people i guess were using the generator and transporting Urqmd and signal all together.

This is feasible (in our case) for moderate statistics (say, 100k events, which you can transport overnight on the batch farm). But now we have to go substantially beyond that (by one or two orders of magnitude), so the issue becomes actual.

Quote: Since it was not used, this way of merging was not maintained and supported by us.

Usage of software always relies on communication and documentation.

Quote: Now according to Mohammad , there are other way to do this merging not directly on Geant data but on the level of Digits which includes time based simulation.

Merging on the digit level is not the proper way. Transport is done independently track by track - there is no interference. But this does not hold for digitisation, which has to take into account the interference of tracks in single channels. Think e.g. of double hits or detector / electronics dead time.

---

---

Subject: Re: Multiple input files / trees

Posted by [Mohammad Al-Turany](#) on Thu, 22 Mar 2012 11:44:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Dear Volker,

First of all there are many features in FairRoot which were not communicated to CBM, even though they could be and they are needed by CBM. This has many reasons, the major one of them that in CBM there is not even a computing session (as in Panda or R3B) where we could go and report about our activities. Also the communication between us (FairRoot core) and CBM developers is not ideal and mostly based on personal mails, but this we do not need to discuss here, and I think this Forum is a first step to correct that.

Now back to your original mail.

The merging we have in FairRoot for now was implemented on the level between Digit and MC and not directly on the digit level. i.e: after simulation of different generator output, the FairRunAna has following methods:

```

/**Set the input signal file
 *@param name :      signal file name
 *@param identifier : Unsigned integer which identify the signal file
 */
void      SetSignalFile(TString name, UInt_t identifier );

/**Set the input background file by name*/
void      SetBackgroundFile(TString name);

/**Add input background file by name*/[COLOR]
void      AddBackgroundFile(TString name);

/**Add signal file to input
 *@param name :      signal file name
 *@param identifier : Unsigned integer which identify the signal file to which this signal
should be added
 */
void      AddSignalFile(TString name, UInt_t identifier );

```

With this you can add several signals to a background file or chain, also signals can be chained, the mixing is controlled by:

Quote: 

```

/**Set the signal to background ratio in event units
 *@param background : Number of background Events for one signal
 *@param Signalid :   Signal file Id, used when adding (setting) the signal file
 * here we just forward the call to the FairRootManager
 */
void BGWindowWidthNo(UInt_t background, UInt_t Signalid);
/**Set the signal to background rate in time units
 *@param background : Time of background Events before one signal
 *@param Signalid :   Signal file Id, used when adding (setting) the signal file
 * here we just forward the call to the FairRootManager
 */
void BGWindowWidthTime(Double_t background, UInt_t Signalid);

```

With these methods you can mix the signals and background but there is no direct pileup here. So do you need to pile up the events on MC level?

if yes then we have to add a merge after we read the signals and background depending on the ratio which one set for there mixing.

best regards,

Mohammad

