Subject: Invalid Events in Prespec Code
Posted by a_boso on Fri, 19 Aug 2016 09:37:23 GMT
View Forum Message <> Reply to Message

Hi everybody!

We noticed that in the 46Cr Coulex part of our analysis (which is the more exotic of the experiment) we have almost 90% of invalid events (events in which the "valid flag" is 0 in all the variables, especially in S4 scintillator). This is not the case in the 46Ti Coulex part where the invalid events where only ~30%.

This is somehow surprising since the beam rates in S4 were:

46Cr
~800 counts per spill (1.2 s)

46Ti
1e5 counts per spill (10s)

If we could recover a situation similar to that of 46Ti it would make a huge difference for the outcome of the analysis.

So I was wondering.. what does "invalid event" mean? How it is decided in the code if an event is valid or not? Is there a way to "relax" this condition and somehow recover some events?

Do you have any idea why we have such a great amount of invalid events?

Thanks!!
Alberto

Subject: Re: Invalid Events in Prespec Code
Posted by Michael Reese on Fri, 19 Aug 2016 13:03:30 GMT
View Forum Message <> Reply to Message

Hi,

The valid flag of any value is set if the value was successfully computed. That happens when the

set_output(NAME_OF_VALUE, 42)

inside any processor is called. That means, if the value has no valid flag set, it was not successfully computed.
Usually, processors are written in a way that they check for the requirements of a computation

```
if (input_valid(NAME_OF_NECESSARY_INPUT_1) &&
input_valid(NAME_OF_NECESSARY_INPUT_2))
{
```

```
  double input1 = input_value(NAME_OF_NECESSARY_INPUT_1);
  double input2 = input_value(NAME_OF_NECESSARY_INPUT_2);
  double result = f(input1,input2);
  set_output(NAME_OF_RESULT, result);
}
```

This guarantees to have only meaningful information propagating along the graph. You can try to track down in which processor the information is missing. With that information I could give a more detailed answer.

In general you can try if it is possible to write a more sophisticated algorithm, such as this:

```
if (input_valid(NAME_OF_NECESSARY_INPUT_1) &&
input_valid(NAME_OF_NECESSARY_INPUT_2))
{
  double input1 = input_value(NAME_OF_NECESSARY_INPUT_1);
  double input2 = input_value(NAME_OF_NECESSARY_INPUT_2);
  double result = f(input1,input2); // f is an algorithm that calculates the result from the two
given numbers
  set_output(NAME_OF_RESULT, result);
}
else if (input_valid(NAME_OF_NECESSARY_INPUT_1))
{
  double input1 = input_value(NAME_OF_NECESSARY_INPUT_1);
  // make a clever computation that needs only one of the values
  double result = f2(input1); // f2 is a sophisticated algorithm that calculates the result only with
one number
  set_output(NAME_OF_RESULT, result);
}
else if (input_valid(NAME_OF_NECESSARY_INPUT_2))
{
  double input2 = input_value(NAME_OF_NECESSARY_INPUT_2);
  // make a clever computation that needs only the other value
  double result = f3(input2); // f3 is a sophisticated algorithm that calculates the result only with
one number
  set_output(NAME_OF_RESULT, result);
}
```

Best regards,
Michael

Subject: Re: Invalid Events in Prespec Code
Posted by a_boso on Fri, 19 Aug 2016 13:45:29 GMT
View Forum Message <> Reply to Message

Hi Michael,

thank you for your fast and precise reply!

I tried to check where the information starts to miss; but it seems it is a common feature of all the variables still from the beginning of the analysis.

For example if I take the processor

```
processor Frs/Scintillators/dEnergySc21  UTILS.Pair
        first <-  FrsCrate.qdc1[16]
        second <- FrsCrate.qdc1[17]
        display first:second
        display first
        display second
end
```

which simply takes the "raw" values from the Frs Crate for Sc21 (but it is the same for Sc41 for example) and plots them I already see that that the
"Frs_Scintillators_dEnergySc21__first_valid" variable is 0 90% of the time.

Since FRS scintillators are the "fastest" detectors we have I did not expect to have such a large amount of invalid events in them.
Moreover it seems that these events are invalid already when they come out from the Frs Crate; this is what I don't understand. What does it mean that raw values from the Crate are invalid?

Thank you very much, and sorry for bothering you!!

Alberto

Subject: Re: Invalid Events in Prespec Code
Posted by Michael Reese on Fri, 19 Aug 2016 19:12:53 GMT
View Forum Message <> Reply to Message

I agree, the Sci41 should have always information in case of trigger 10,9,8,7,6.
Perhaps there was a problem with the QDC?
You can try to get the Sci41 information from a different module.
It should be also in the first multihit-TDC in the LyccaTargetTofCrate:

LyccaTargetTofCrate.mhtdc0[20] #Sc41L
LyccaTargetTofCrate.mhtdc0[22] #Sc41R

This has to be put into a multihit preprocessor (perhaps you find it already preprocessed somwhere) before being used.

Best regards,
Michael

Subject: Re: Invalid Events in Prespec Code
Posted by Michael Reese on Sat, 20 Aug 2016 07:54:38 GMT
View Forum Message <> Reply to Message

Actually, there might be a problem with the unpacker, too.
I'll have a look.

Subject: Re: Invalid Events in Prespec Code
Posted by Michael Reese on Mon, 22 Aug 2016 09:14:36 GMT
View Forum Message <> Reply to Message

Hi Alberto,

can you please tell me one lmd filename where this problem occurs (one of the Cr runs) and one lmd file where it is better (one of the Ti runs). I would like to see if there is a difference in the raw data from the FRS crate in those files.

Best regards
Michael

Subject: Re: Invalid Events in Prespec Code
Posted by a_boso on Mon, 22 Aug 2016 13:24:27 GMT
View Forum Message <> Reply to Message

Hi Michael,

this feature is there in all the runs; see for example

46Ti_coulex_AR12_0030.lmd
and
46Cr_coulex_AR16_0061.lmd

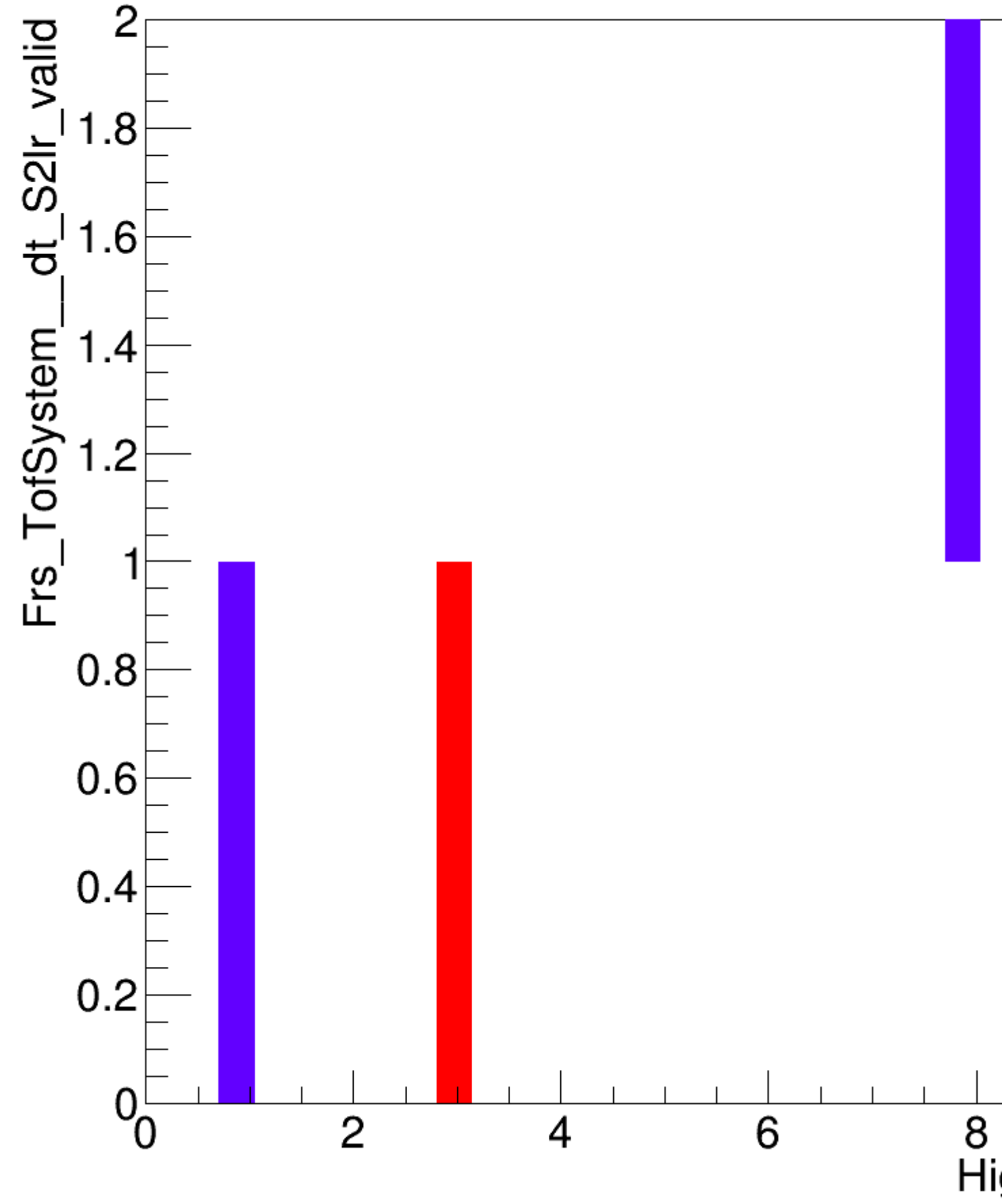
However, following your last reply to this post I plotted one scintillator variable valid flag versus the trigger value, and I obtained the behaviour reported in the attachment.

So it looks like we have a huge amount (~90%) of trigger 3 events, so AGATA alone. This is not the case for 46Ti, and looks a little bit strange to me. These events are probably not usable for the cross section measurement; however it seems that there are no mistakes in the analysis and this is really what we got from the experiment!!

Thank you!!
Alberto

File Attachments

1) Screenshot from 2016-08-22 09:47:28.png, downloaded 688 times


Frs_TofSystem__dt_S2lr_valid:HighLev
Frs_TofSystem__dt_S2lr_valid
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
Hi