

---

Subject: prespec git updates

Posted by [miree](#) on Mon, 11 Aug 2014 17:51:44 GMT

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

---

I've done some slight changes (updates, correction of mistakes) to the prespec-tutorial that you can find in the prespec/doc/PreSPEC\_Tutorial directory of the package. Some while ago, I also added a FAQ section, describing things like: making logical AND/OR of different conditions and how to define and use gates.

Michael

---

---

Subject: Re: prespec git updates

Posted by [mlcortes](#) on Thu, 14 Aug 2014 11:43:33 GMT

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

---

I pushed the following changes in Frs Id processor

- Calculation of AoQ without correcting the Rho included
  - A new way to get charged states based on P. Mayet PhD thesis and J. Benlliure, Nucl. Phys A660 87 (1999)
  - A parameter to indicate if there is S1 degrader in the settings. This is useful when calculating the BRho of the first half of FRS and charged states
- 

---

Subject: Re: prespec git updates

Posted by [mlcortes](#) on Mon, 18 Aug 2014 09:21:20 GMT

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

---

I have pushed a change in the position correction of the musics.

Instead of

$dE = dE + x * value / dE$

i take

$dE = dE - x * value$

Where value is the slope of the line one fits to the  $dE_{geo}$  vs position in the musics plot.

---

---

Subject: Re: prespec git updates

Posted by [miree](#) on Tue, 19 Aug 2014 15:18:37 GMT

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

---

Added two small programs to the repository.

They extract the first and last GTS timestamp in an lmd file (written by MBS system) or from

the set of all psa\_\*.adf files in a NARVAL run folder. These tools can be used to verify if you're pairing the correct lmd file with a given NARVAL run.

they are located in the prespec/tools subdirectory and are installed when typing "make install" together with the preplay program.

use like this (for lmd files)

```
gts_timestamps_mbs /d/rising02/feb_14/data/Fe56_coulex_ar3_0106.lmd > /dev/null
```

outputs:

```
/d/.../Fe56_coulex_ar3_0106.lmd < Mon Feb 24 07:12:56 2014> 23270254723987  
23375015876804
```

use like this (for adf files)

```
gts_timestamps_adf  
/SAT/hera/gamma/AGATAdata/040414_bentley/run_0011_46Ti_coulex/Data/
```

outputs:

```
/SAT/..Data/ <Sat Apr 5 03:50:07 2014 > 5630037697086  
7425940419075
```

---

The first/last number is the first/last GTS timestamp in the file/folder.

Yes, these numbers are large! They count the number of 10ns GTS time-ticks since the start of the GTS system.

By looking at them you can see if time ranges of two event sources do overlap or not.

---

---

**Subject: Re: prespec git updates**  
Posted by [miree](#) on Fri, 14 Nov 2014 13:53:46 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Recent, notable additions to the prespec package:

- 1) A new preplay option to change the name of the TTree in the root file
  - 2) simplification of the MGT tracking processor
  - 3) some new processors in the UTILS plugin (see git for details)
- 

---

**Subject: Re: prespec git updates**  
Posted by [miree](#) on Thu, 27 Nov 2014 20:26:52 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

There is a new processor for the mutual alignment of correlated signals. It can determine a set of offsets, such that all differences between pairs of signals will be zero.

One possible application of this: Consider the time signals of a LYCCA ToF detector. These signals have an offset with respect to each other, because of slightly different cable length (and maybe other reasons). As a consequence, if a particle hits the detector, Then signal from PMT\_a will come not exactly at the same time as the signal from PMT\_b, even if both signals were induced by the same particle. This offset is of course the same for each event. The processor CircularMembraneScintillator takes these offsets into account. But in practice, these offsets tend to induce difficulties in the calibration. Therefore, it might make sense to align the time signals before feeding them into the final processing.

```
processor ToFstart/SignalAlignment UTILS.InterSignalAlignment
```

```
  INPUT[0:15]  <-  LyccaTargetTofCrate.mhtdc0[0:15]
```

```
  INPUT[16:31]  <-  LyccaTargetTofCrate.mhtdc1[16:31]
```

```
  display INPUT_DIFFERENCES 10000,-5000,5000
```

```
  display OUTPUT_DIFFERENCES 10000,-5000,5000
```

```
end
```

```
# use output of the time alignment instead of the raw data
```

```
processor Lycca/ToFStart/Preproc  UTILS.MhTdcPreprocessor
```

```
  input[0:31]  <-  ToFstart/SignalAlignment.OUTPUT[0:31]
```

```
end
```

The parameter file for the InterSignalAlignment might look like this:

NUM_SIGNALS	32
NUM_BINS	8000
BIN_WIDTH	1
CALIBRATE	1
WRITE_INTERVAL	1000

Where the most important parameter is NUM\_SIGNALS (32 PMTs in this case). The other parameters are for tuning the automatic determination of the offsets. Look into the header file for detailed description.

I've tried out a new style of writing the processor code: I wrote every in/output,parameter,etc. in CAPITAL letters. The reasoning behind this: Trying to avoid the common mistake, that one accidentally writes num\_signals when actually parameter(num\_signals) would be correct. The capitals are clearly enums and no normal variables.

---

Subject: Re: prespec git updates  
Posted by [miree](#) on Fri, 12 Dec 2014 12:39:39 GMT

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

---

Hello,

A severe problem in the so called "Core Tracking" algorithm was discovered by Liliana (the name of the processor is "GammaTrackingCore" in the AGATA plugin) .

First, the algorithm was throwing out all events with only one PSA hit the full array. That reduces the statistics.

Second, it produced one additional "nonsense gamma". The second point increases the gamma multiplicity by one, but does not affect the gamma spectra.

I have rewritten the algorithm and copied the old one to "GammaTrackingCoreWithBug", in case you want to quantify the losses you had in your spectra. The changes are already pushed to the git repository.

I did the comparison for one of the 80Kr Coulex runs (everything else in the analysis is identical).

The counts in the 80Kr peak (at 616 keV):

old: 569.95 +- 45.1105

new: 688.684 +- 48.2208

(the histograms are background subtracted. That is why some of the bins may contain fewer counts after correcting the mistake.)

So 20% of the counts were lost because of that mistake.

I don't know if under different conditions (higher/lower rate, heavier particles, etc.) the loss is bigger or smaller.

I'm sorry for that mistake!

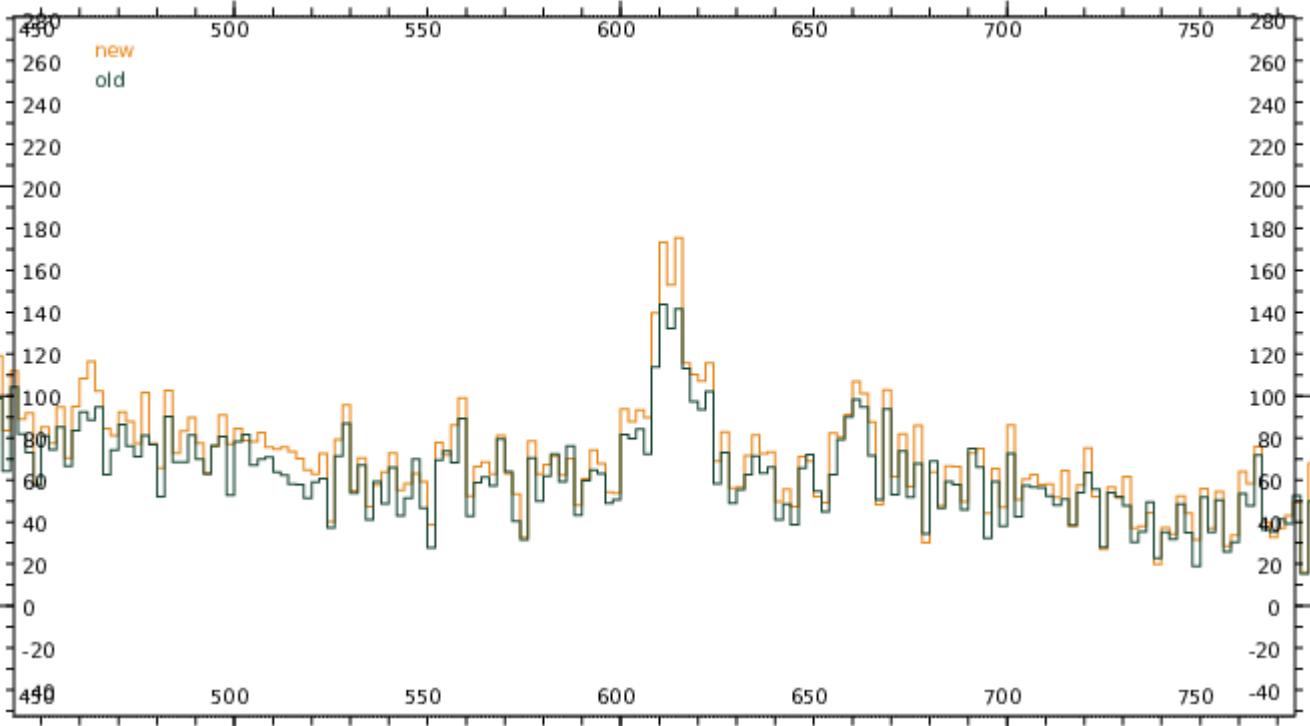
Michael

---

#### File Attachments

1) [old\\_new.png](#) , downloaded 922 times

---




---

Subject: Re: prespec git updates

Posted by [miree](#) on Wed, 21 Jan 2015 10:49:14 GMT

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

---

Fixed the uncorrected A/Q output "AoQ\_uncorrected" of FRS Id Processor.  
 (The uncorrected A/Q does not depend on the x-coordinates at S2 and S4.)

---



---

Subject: Re: prespec git updates

Posted by [mlcortes](#) on Thu, 22 Jan 2015 13:42:56 GMT

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

---

I included a new processor to get the ToF from the MhTDCs. Is called TofSystemMhTDCSc and is in the UTILS plugin.  
 It calculates differences between hits for ToF left and Tof right and takes the first combination which gives ToF inside a window.  
 Using this hits time left - right for each scintillator is also given

---



---

Subject: Re: prespec git updates

Posted by [Michael Reese](#) on Mon, 01 Aug 2016 11:35:20 GMT

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

---

There is a new feature in the config file parser.

It can evaluate integer expressions now.

If it finds curly braces, it will evaluate the expression inside and replace the complete string (including the curly braces) with the result of that expression. It supports common operator

priority ('^' binds more than '\*' or '/' which bind more than '+' and '-') and parentheses '(' and ')'.  
Example:

```
"{2+2*3}" -> "8"  
"{{2+2)*3}" -> "12"
```

It is in particular meant to be used in for loops, for example:

```
processor even_channels UTILS.SingleArray  
  for $i in [0:15]  
    array[$i] <- AdcCrate.adc[{2*$i}]  
  end  
end
```

```
processor odd_channels UTILS.SingleArray  
  for $i in [0:15]  
    array[$i] <- AdcCrate.adc[{2*$i+1}]  
  end  
end
```

---