Subject: Re: problem with a processor (software-wise)
Posted by thuyuk on Wed, 13 Aug 2014 10:32:37 GMT
View Forum Message <> Reply to Message

Hi Michael,

Thank you for the explanation on how arrays work in the code, although I read this part of the tutorial before, it's useful to have it here if somebody wants to get it easier.

I searched through the available processors in plugins/UTILS, but, I couldn't find one or a combination of them to do what I want to do.

So, I followed your suggestion, and created another processor. Please find below how it looks like:

This the cpp:

```
#include "ArraysWithCondition.hpp"
#include <iostream>
#include <cmath>

ArraysWithCondition::ArraysWithCondition(const std::string &config_dir,
        const std::string &name,
        const bool no_files)
  : Processor(name, config_dir, no_files, std::vector<int>())
{
  // make all the in/outputs, parameters and coefficients
  // known by name.


  NAME_INPUT_ARRAY(value1, 7);
  NAME_INPUT_ARRAY(value2, 7);

  NAME_OUTPUT_ARRAY(GatedValue, 7);

  NAME_CONDITION_ARRAY(gate, Window1D, 7);

  init();
  read_conditions();
}

ArraysWithCondition::~ArraysWithCondition()
{
}

void ArraysWithCondition::process(prespec::viscon::Interface &viscon_interface, int trigger)
{


 for(int i = 0; i < input_array_size(value1); i++)
 {
  int index = input_array_index(value1, i);
```

```
  double val2 = input_array_value(value2, index);
  double val1 = input_array_value(value1, index);
  double left_gate = condition_array(gate, index, 0);
  double right_gate = condition_array(gate, index, 1);
  if(val2 < right_gate && val2 > left_gate)
  {
   fill_output_array(GatedValue, index, val1);

  }

 }



}
```

and the hpp:

```
#ifndef UTILS_ArraysWithCondition_HPP_
#define UTILS_ArraysWithCondition_HPP_

#include <prespec/process/Processor.hpp>


//! @brief A processor that evaluates the information coming from a
//!        thin circular plastic membrane detectors, surrounded by
//!        many photomultipliers. These detectors are used in the
//!        Lycca time of flight subsystem.  WTF!!!!! wrong doc!
//!
//! It evaluates the time of the particle impact on the membrane, given
//! the particle position (deduced by other tracking detectors).
//! If this information is missing, the processor tries to deduce the
//! impact position from the time information. That is possible with an
//! accuracy of between (6-10 milimeters).
class ArraysWithCondition : public prespec::process::Processor
{
public:
  ArraysWithCondition(const std::string &config_dir,
  const std::string &name,
  const bool no_files = false);
  ~ArraysWithCondition();

  virtual void process(prespec::viscon::Interface &viscon_interface, int trigger);

  double analyze(double x_particle, double y_particle, bool find_position = false);

  // the inputs to this processor
  enum InputArray
    {
      value1,
```

```
      value2,
    };

  // the outputs of this processor
  enum OutputArray
    {
      GatedValue,
    };


   enum Conditions
    {
      gate,
    };

};


#endif
```

This the part in Store.config:

```
processor HighLevel/particle/incoming/MusicPileUp UTILS.ArraysWithCondition
 value1[0:6] <- MuiscPileupCor.music[0:6]
 value2[0:6] <- MuiscPileupCor.music[8:14]

 display GatedValue
end
```

And this is how the condition file looks like:

```
gate[0]  25 160
gate[1]  25 160
gate[2]  25 160
gate[3]  25 160
gate[4]  25 160
gate[5]  25 160
gate[6]  25 160
```

Well, with these files, the compilation is done without errors. But when I want to run this processor, I get this:

```
prespec/process/Processor.cpp:665: const double&
prespec::process::Processor::input_array_value(int, int): Assertion
`input_arrays_[channel].size() > i' failed.
```

First of all, I looked through the written processors, and couldn't find one to process the MUSIC

Pile up data. Second, I have no idea how to process these data, due to the high counting rate, we possibly suffer because of the pile-up in MUSIC, which makes noise in the higher Z region in the ion selection in FRS. So this code intents to put a 1-D gate on the peaks that appear on the spectra of the last 8 channels of the MuiscPileUp Crate, and accept the data from the first 8 channels with this condition.

Maybe I'm wrong, maybe this is not the way to process these data, this is another subject to discuss, but I don't see the reason why the code fails with this kind of error message. The array size is 7 in this case, because two of the channels are missing in the crate. I suppose to fill 7 indices of the input arrays, but why I cannot ask the input value inside any of them?
