

## dunetpc - Feature #12701

### New module and services for raw data preparation

05/18/2016 08:13 AM - David Adams

<b>Status:</b>	Closed	<b>Start date:</b>	05/18/2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	David Adams	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Description</b>			
<p>After unpacking into larsoft RawDigit format, there are many actions taken to prepare the raw data for reconstruction starting with cluster finding. These actions include</p> <ul style="list-style-type: none"><li>• conversion from int to float</li><li>• stuck bit mitigation</li><li>• coherent and incoherent noise removal</li><li>• pedestal addition</li><li>• deconvolution</li><li>• ROI building</li></ul> <p>At present these are done within separate modules for FD and 35t with a service used for the deconvolution. Three separate modules are called for 35-ton with a copy of the (large) raw data written out for each. We would like to have a common module to avoid the code duplication and a single module to avoid storing multiple copies of the data. It is probably desirable to have the option of no module, i.e. to do the data preparation in the same module as the cluster finding and avoid storing even one extra copy of the data.</p> <p>The proposal here is to move the code for the above actions into services (then to tools when available). These services would all have interfaces so that alternative implementations could be plugged in at run time (by changing FCL) with no change in the C++ client code. A module with appropriate calls to these services (i.e. service interfaces) will be added so that a copy of the prepared raw data and be written in the event record. To meet the last requirement, this sequence of service/tool calls will be but into a high-level service and the module would simply call that service.</p> <p>Except for the last step (which probably should be added), this is what was done for detector simulation as discussed in <a href="#">#11777</a>.</p> <p>This has been discussed (and agreed?) in many DUNE FD and 35-ton sim/reco meetings. Should I proceed with this or do we first want more discussion?</p>			
<b>Related issues:</b>			
Blocked by dunetpc - Bug #12802: Create service to convert int raw data to fl...		<b>Closed</b>	<b>05/27/2016</b>
Blocked by dunetpc - Feature #12906: Create service to remove coherent noise ...		<b>Closed</b>	<b>02/18/2016</b> <b>02/18/2016</b>
Blocked by dunetpc - Feature #11750: Interface for noise subtraction service		<b>Closed</b>	<b>02/17/2016</b>
Blocked by dunetpc - Bug #13306: Clarify range for ROI in recob::Wire		<b>Closed</b>	<b>07/20/2016</b>
Blocked by dunetpc - Bug #13550: New prep data Wire output has different size...		<b>Closed</b>	<b>08/15/2016</b>

### History

#### #1 - 05/27/2016 08:22 AM - David Adams

- Status changed from New to Assigned
- Assignee set to David Adams

There were no objections and I have need for some of the tools described here and so I am proceeding with this.

The new data preparation service will call a series services (later tools) via service interfaces. The first of these converts the raw int signals to float, identifies underflows, overflows and stuck bits, and subtracts pedestals. It is the topic of issue [#12802](#).

#### #2 - 05/27/2016 08:23 AM - David Adams

- Blocked by Bug #12802: Create service to convert int raw data to float, flag underflows, overflows and stuck bits, and subtract pedestals added

#### #3 - 06/10/2016 07:17 AM - David Adams

- Blocked by Feature #12906: Create service to remove coherent noise from ADC data added

**#4 - 06/10/2016 07:17 AM - David Adams**

- Blocked by Feature #11750: Interface for noise subtraction service added

**#5 - 07/18/2016 01:11 PM - David Adams**

I gave a talk summarizes the status of this issue at today's FD reco meeting: <https://indico.fnal.gov/conferenceDisplay.py?confId=12518>.

**#6 - 07/20/2016 10:26 AM - David Adams**

- Blocked by Bug #13306: Clarify range for ROI in recob::Wire added

**#7 - 07/20/2016 10:27 AM - David Adams**

I have added a service to build ROIs and am now working on service to build recob::Wire. A pointer to the latter is added to AdcChannelData. There is some ambiguity in the ROI definition that I discuss in issue [#13306](#).

**#8 - 07/26/2016 08:19 AM - David Adams**

StandardRawDigitPrepService is now complete. It can take in digits and put out wires doing all the steps in the current 35-ton data preparation module. I still need to add a module wrapper for this service. I would also like to reorganize so that the signal finding and ROI building can be done with same interface. Then there is work to do validation for 35-ton and then updated (if needed) for FD and protoDUNE and do validation there.

**#9 - 08/03/2016 07:31 AM - David Adams**

There is now a module wrapper for data prep: DataPrepModule and configuration for that module at

```
dunetpc/fcl/dune35t/reco/new_standard_reco_dune35tdata.fcl
```

I am trying to validate this by comparison with

```
dunetpc/fcl/dune35t/reco/standard_reco_dune35tdata.fcl
```

Is this the correct reference, i.e. what we use for production?

I am seeing large discrepancy between the new module/configuration and this reference.

**#10 - 08/03/2016 11:38 AM - David Adams**

I see Tingjun has dropped services.user from the old 35t production fcl. I do the same for the new.

**#11 - 08/04/2016 10:59 AM - David Adams**

I presented the status of this task at yesterday's 35-ton sim/reco meeting.

Much of the discrepancy between new and old data prep shown at that meeting was due to a defect where the deconvolution was not recorded. I fixed this and the new spectra are much closer to the old. The biggest difference is the presence of apparently noisy channels in the new reco. I will add the option to skip known bad channels.

**#12 - 08/04/2016 11:27 AM - David Adams**

All:

I am again unsure about the convention used to record bad channels. I thought we use online channel numbers, but this code from CalWireDUNE35t\_module.cc:

```
art::Ptr<raw::RawDigit> digitVec(digitVecHandle, rdIter);
channel = digitVec->Channel();

// skip bad channels
if(!chanFilt->BadChannel(channel)) {
```

suggests the channel status provider should be called with offline channel numbers.

Should I use online or offline channel numbers in calls to ChannelStatusProvider?

Thank you.

**#13 - 08/04/2016 11:51 AM - Tingjun Yang**

Hi David,

The bad channel map uses offline channel numbers.

Tingjun  
David Adams wrote:

All:

I am again unsure about the convention used to record bad channels. I thought we use online channel numbers, but this code from CalWireDUNE35t\_module.cc:

[...]

suggests the channel status provider should be called with offline channel numbers.

Should I use online or offline channel numbers in calls to ChannelStatusProvider?

Thank you.

**#14 - 08/04/2016 12:43 PM - David Adams**

Thanks Tingjun. I see Michelle told me the same in May. I was probably confusing this with the convention for storing pedestals.

With that change, the agreement between new and old reco improves, There are a few places where the old and new differ by a few fC. I suspect this is the deconvolution magnifying the ADC count rounding differences we expect between the new and old reco. The new reco stores the stuck bit interpolation and noise removal results as floating while the old used integer.

Can someone remind me of the ADC to fC conversion factor here?

**#15 - 08/04/2016 01:39 PM - Tingjun Yang**

There is the parameter we use in signal shaping service  
ADCPerPCAtLowestASICGain: 13160 #ADC/pC (2.8 ADC/mV \* 4.7 mV/fC \* 1000)  
This is for the lowest gain 4.7 mV/fC. You need to scale it for other gains.

Tingjun  
David Adams wrote:

Thanks Tingjun. I see Michelle told me the same in May. I was probably confusing this with the convention for storing pedestals.

With that change, the agreement between new and old reco improves, There are a few places where the old and new differ by a few fC. I suspect this is the deconvolution magnifying the ADC count rounding differences we expect between the new and old reco. The new reco stores the stuck bit interpolation and noise removal results as floating while the old used integer.

Can someone remind me of the ADC to fC conversion factor here?

**#16 - 08/09/2016 07:52 AM - David Adams**

Tingjun:

In the 35-ton simulation reconstruction (standard\_reco\_dune35tsim.fcl), I see:

```
ADCPerPCAtLowestASICGain: 11700
ASICGainInMVPerFC: [7.8, 7.8, 7.8]
```

Does this mean the gain there is as follows?:

$$G = 11.7 * 7.8 / 4.7 = 19.4 \text{ ADC/fC} ?$$

Thanks.  
David

**#17 - 08/09/2016 11:10 AM - David Adams**

I had private communication from Michelle that the ADC response is 2.808 ADC/mV. I confirmed this with Matt Worcester who also indicated that the ADC response and ASIC gain (4.7 mV/fC lowest setting) are measured independently. He was not sure of the uncertainty in these numbers but thought the difference with 2.8 might be significant. Given the "best" values the FCL parameter should be set to

```
ADCPerPCAtLowestASICGain: 13198
```

i.e. 13.2 ADC/fC.

The 35t simulation uses log-gain ADC response 11700 ADC/pC with ASIC gain 7.8 mV/fC and so the overall gain in both simulation and

reconstruction is  $11.7 \times 7.8 / 4.7 = 19.4$  ADC/fC.

**#18 - 08/09/2016 12:33 PM - David Adams**

Tingjun:

I having problems running (old) 35-ton reco in v06\_02\_00:

```
terminate called after throwing an instance of 'cet::coded_exception<fhicl::error, &fhicl::detail::translate>'
  what(): ---- Parse error BEGIN
  Local lookup error
  ---- Can't find key BEGIN
    dune35tdata_emshower (at part "dune35tdata_emshower")
  ---- Can't find key END
  at line 95, character 26, of file "/home/dladams/dudev/dudev06/workdir/localProducts_larsoft_v06_02_00_e10_p
rof/dunetpc/v06_02_00/job/standard_reco_dune35tdata.fcl"
  included from line 6 of file "./oldrecowire_35tdata.fcl"

  emshower:                @local::dune35tdata_emshower
                        ^
---- Parse error END
```

I don't see a defn of dune35tdata\_emshower anywhere in dunetpc.

da

**#19 - 08/09/2016 12:37 PM - David Adams**

I have added Michael Wallbank to this ticket. He may be able to tell us where to find dune35tdata\_emshower.

**#20 - 08/09/2016 12:47 PM - Michael Wallbank**

dune35tdata\_emshower is configured in larreco/larreco/ShowerFinder/job/showerfindermodules\_dune.fcl. I only added it a couple of days ago so I guess it's possible not everything has fit together perfectly... I'll take a look to make sure it's picked up correctly.

**#21 - 08/09/2016 12:58 PM - Michael Wallbank**

I've just tried running the reconstruction myself and it works fine. I guess it needs the latest version of larreco -- since I pushed it since the last release, if you don't have your own larreco kept up-to-date then it'll take until the next release for it to be picked. Sorry, I should have considered this. You could just comment it out for the moment if needed to.

**#22 - 08/09/2016 01:07 PM - David Adams**

I have commented out the emshower references in standard\_reco\_dune35tdata.fcl and new\_standard\_reco\_dune35tdata.fcl. we should uncomment those before we make the next release. Note that the latter is intended to (soon) replace the former and so any additional changes made in the former should also be made in the latter.

**#23 - 08/09/2016 01:16 PM - David Adams**

I have checked out larreco and once I confirm I can run reco after uncommenting my changes above, I will commit that, i.e. put things back to how they were. I will probably do this tomorrow.

**#24 - 08/10/2016 08:18 AM - David Adams**

I have confirmed I can run with the head of larreco and have restored emshower reco for dune 35t data.

**#25 - 08/10/2016 08:30 AM - David Adams**

Tingjun:

I would like to convert our standard 35-ton production to use the new data prep. I will show some results at the the 35-ton meeting today.

I see the following fcl files in dunetpc/fcl/dune35t/reco reference caldata:

```
emhits.fcl
reco_dune35t_blur.fcl
standard_reco_dune35tdata_fasthit.fcl
standard_reco_dune35tdata.fcl
standard_reco_dune35t.fcl
```

Should I modify all of these, i.e. do we use them all for production?

Temporarily, I have prefixed the modified file names with "new\_" and these two are modified and included in dunetpc:

new\_standard\_reco\_dune35tdata.fcl  
new\_standard\_reco\_dune35tsim.fcl

Note that I also propose to rename standard\_reco\_dune35t.fcl --> standard\_reco\_dune35tsim.fcl.

**#26 - 08/10/2016 09:26 AM - David Adams**

I see that standard\_reco\_dune35tdata\_fasthit.fcl uses Sigmoidfilter which is another producer that reads and write RawDigit. However its output is not explicitly referenced by any other producers in the fcl file. I have added Karl to this ticket as it appears he is the author of Sigmoidfilter. I would like to convert Sigmoidfilter from a module to a service and include it in data prep if it is to be part of standard reconstruction.

I also see that standard\_reco\_dune35tdata\_fasthit.fcl calls RawHitFinder which creates hits directly from RawDigit. It reads an intermediate digit collection (after unstick but before daq and sigmoid). I can easily modify data prep to provide the option to write out intermediate states either in the form of digits or wires if we want this capability. Note one of the goals of data prep is to avoid writing all the intermediate states.

I have defined data prep to read raw::RawDigit and end with recob::Wire. It may be that we should instead have it end with recob::Hit and make it optional to write out intermediate digits and wires. If data prep include the hit finder(s), then there could be no need to write wires or any intermediate digit/wire collections.

**#27 - 08/10/2016 12:49 PM - David Adams**

I showed data and sim validation results at today's 35-ton sim/reco meeting. We agree that I should proceed with updating the 35-ton production fcl file to use the new DataPrep service. I have also renamed some of the files to improve consistency. Here are the old and new names:

Old name	New name
standard_reco_dune35t.fcl	standard_reco_dune35tsim.fcl
standard_reco_dune35tdata.fcl	standard_reco_dune35tdata.fcl
standard_reco_dune35t_cc.fcl	standard_reco_dune35tsim_cc.fcl
standard_reco_dune35t_milliblock.fcl	standard_reco_dune35t_milliblock.fcl
standard_reco_dune35tdata_fasthit.fcl	standard_reco_dune35tdata_fasthit.fcl
reco_dune35t_blur.fcl	reco_dune35tsim_blur.fcl
emhits.fcl	reco_dune35tsim_emhits.fcl

The first two are the ones for which results were shown today. For the remainder, I verified the old and new Wire outputs are the same with one part in 1.e-6.

The following file standard\_reco\_dune35t\_cc.fcl is so-far unchanged. It is the topic of comment #26.

**#28 - 08/10/2016 09:46 PM - Thomas Warburton**

Hi David,

The sigmoid filter module applies the high low pass (sigmoid) filter which is done in cal data. It was written as a separate module so that people using fast hit could choose whether or not they apply it. I didn't want to force anyone to do so hence it not being referenced further down the fcl file.

In the process of making the data prep a series of services it should be made into a service, so I am happy with this. You need to be careful that it is not applied if cal data is later applied, as then the filter would be applied twice which would mean performing two sets of FFTs which will definitely not preserve the RawData.

You also need to be careful in the fact that fast hit is ran on the RawData whereas GausHit is ran on wires. This may mean it makes more sense to define it as ending with hits? Otherwise there would have to be two different methods depending on whether you run fast hit or gaushit.

**#29 - 08/12/2016 08:37 AM - David Adams**

Karl:

Thanks for you comments. I agree we should move the sigmoid code to a noise reduction service and have created a new issue for this: [#13523](#).

I am not sure I follow you comment "preserve the RawData". In the new model, the raw data is never modified or event copied. Instead a the results of data preparation are written to the recob::Wire container.

If we are going to run multiple filters that use the (Fourier) transformed data representation, that it might make sense to run this transform once and record along with the standard representation. It would be natural to include this in the transient data AdcChannelData.

Whether or not we add hit finding to data prep, I would like to see the fast hit code moved to a service/tool that takes AdcChannelData as input so it does not have to repeat the uncompression and pedestal subtraction and we do not have to write out the mitigated (i.e. sticky-bit interpolated) raw data. Let me know if you agree and I will create an issue for that.

**#30 - 08/12/2016 06:19 PM - Thomas Warburton**

When we the separation of the sigmoid filter was originally discussed Michelle noted that we did not want to apply fourier transforms on the data multiple as 'it would not preserve the data' were her words I believe. Essentially what she meant is that by doing multiple FFTs on the data we could inadvertently change the data we are looking at.

I did not mean to insinuate that we were in some way changing the physical copy of the raw data as that is just wrong, I merely meant that we may introduce things into the data which were not there originally, hence changing the data.

I don't think I am qualified to say what should be done with the fast hit finder as I have not had any part it's construction or maintenance. I was merely pointing out something which needs to be considered in order for the fast hit code to remain functional.

**#31 - 08/15/2016 08:55 AM - David Adams**

- Blocked by Bug #13550: New prep data Wire output has different size than old caldata added

**#32 - 08/15/2016 08:56 AM - David Adams**

Vito reports the Wire data size is different with the new reco. Issue [#13550](#) covers that.

**#33 - 09/07/2016 09:03 AM - David Adams**

- Status changed from Assigned to Closed

The code for the new data prep is in place and the production fcl is updated for 35-ton and (single-phase) FD. I have opened issues to do the same for protoDUNE ([#13781](#)) and dual-phase ([#13782](#)). While some issues such as [#13523](#), the dataprep seems to be working well and I close this issue.