

# Architecture review status report

## Core service review

Gianluca Petrillo

Fermi National Accelerator Laboratory

LArSoft Coordinators' Meeting, December 8<sup>th</sup>, 2015



This is a short update on top of [the one from ~~one~~three weeks ago](#):

- renamed also two “non-core” services:
  - `IChannelStatusService`  $\Rightarrow$  `ChannelStatusService`
  - `IDetPedestalService`  $\Rightarrow$  `DetPedestalService`

Their providers have lost their `I` too, but not the `Provider`<sup>1</sup>;  
everything still in `lariov` namespace

- written legacy `LArPropertiesServiceArgoNeuT` and `DetectorPropertiesServiceArgoNeuT`, intended for use by `ArgoNeuT`

---

<sup>1</sup>Both `lariov::ChannelStatus` and `lariov::DetPedestal` classes are already defined and the standard naming pattern would conflict with them.

This is a short update on top of the one [from one week ago](#):

- added FHiCL *configuration validation* for `LArPropertiesServiceStandard` and `DetectorPropertiesServiceStandard`
- put together a script to help with the code changes;  
**it will not be sufficient! manual checks and edits are still needed**
- updated [the wiki page](#); now mostly stable

# Temperature is changing

We agreed to **move** `Temperature()` **method** from `LArProperties` to `DetectorProperties`.

This dragged also:

- `Density()`
- `Eloss()`
- `ElossVar()`
- `BirksCorrection()`
- `ModBoxCorrection()`
- Sternheimer parametrization

Temperature and Sternheimer parameters need to be moved from `LArProperties` to `DetectorPropertiesServiceStandard` configuration.

The to-do list from two weeks ago is short:

- remove `DatabaseUtil` from where it's not needed (can wait)
- verify and update all `art::ServiceHandle` uses and service dependences (can wait)
- fix the design issue with `DetectorClocksStandard` (can wait)  
*first try: failed!*
- write more unit tests (some already written)

I declare “it should do”...

# Quick summary of changes

Changes include:

- access to services changes syntax
- some services and providers change name
- some functions change provider
- some parameters change service configuration

That's a lot to swallow...

More details in the aforementioned [wiki page](#):

[https://cdcvms.fnal.gov/redmine/projects/larsoft/wiki/Core\\_Services\\_Review](https://cdcvms.fnal.gov/redmine/projects/larsoft/wiki/Core_Services_Review)

Problems?

Just ask (mailing list, personal mail, drop by).

# Short-term plans

- user code is updated!
  - MicroBooNE
  - LArIAT
  - ArgoNeuT
  - SBND

DUNE code might be behind — I think Jonathan will be back on this soon

- **delivery with the invitation to test it**
- **code (all updated to LArSoft v04\_30\_00):**

```
LArSoft feature/jpaley_LArPropertiesBreakup
argoneutcode feature/gp_ServiceCoreReview
lar1ndcode feature/gp_ServiceCoreReview
lariatsoft feature/gp_ServiceCoreReview
uboonecode feature/gp_ServiceCoreReview
```

Soon we'll have `v05_00_00_rc` branches; there is where the final testing takes place (see today's LArSoft status talk).

# Backup



# Dismantling DatabaseUtil

DatabaseUtil service contains two classes of methods:

- 1 direct access to a database, only used by ArgoNeuT
- 2 MicroBooNE specific database access code

What's going to happen:

- 1 a legacy version of core properties will be created for use to ArgoNeuT
- 2 MicroBooNE specific code will be moved to `uboonecode`
- 3 each service in need will implement access to a specific DB
- 4 recommended access is not direct but via `libwda`
- 5 if a common pattern will emerge, common functions can be collected and coagulated into a service

## Design choices:

- core service providers assume they are always up to date
- users should also assume that
- the framework is responsible to update them as needed

## Funny facts:

- `DetectorClocksStandard` providers information based on a data product (`raw::Trigger` collection)
- that data product might have been created in this job, and `DetectorClocksStandard` would not know
- MicroBooNE code explicitly forced `TimeService` to check
- such a command is not available in `DetectorClocks` interface

Trying to force automatic service update on each event before each module (`sPreModule` action) did not work: *art* does not support reading from the event at that time (may be hackable...).