
Status of LArSoft Services Factorization

LArSoft Stakeholders Meeting

Nov. 3, 2015

Recap: Major/Breaking Changes + Some Notes

- `TimeService` now renamed to `DetectorClocks`
- `SimpleTimeService` no longer exists (use `DetectorClocks`, not `DetectorClocksService`)
- Methods that are truly detector-specific have been moved out of `LArProperties` and into `DetectorProperties` (eg, `Efield` and dependencies). Impacts calculations of drift velocity, Birks suppression, etc.
- `Service handles` no longer used to access data. The service handles provide a const ptr to a data provider that can be used to access data. Eg:

```
const dataprov::DetectorProperties* detprop =  
    art::ServiceHandle<util::DetectorPropertiesService>()->getDetectorProperties();  
detprop->DriftVelocity();
```

Recap: Major/Breaking Changes + Some Notes

- `TimeService` now renamed to `DetectorClocks`
- `SimpleTimeService` no longer exists (use `DetectorClocks`, not `DetectorClocksService`)
- Methods that are truly detector-specific have been moved out of `LArProperties` and into `DetectorProperties` (eg, `Efield` and dependencies). Impacts calculations of drift velocity, Birks suppression, etc.
- `Service handles` no longer used to access data. The service handles provide a const ptr to a data provider that can be used to access data. Eg:

```
const dataprov::DetectorProperties* detprop =  
    art::ServiceHandle<util::DetectorPropertiesService>()->getDetectorProperties();  
detprop->DriftVelocity();
```

No one liked this, especially me...

Proposed Change

- After extensive consultation/discussion with Gianluca and Marc, we have come up with the following:

```
auto const *detprop = lar::providerFrom<util::IDetectorPropertiesService>();
```

- Note:
 - `lar::provider` is a simple templated function that returns a const ptr to the data provider object. Lives in `larcore/CoreUtils`
 - `util::IDetectorPropertiesService` is a pure virtual `art::Service` that has a `IDetectorPropertiesService::provider()` method that returns a const ptr to the data provider object.
 - `auto` can of course also be written as `dataprov::IDetectorProperties`, which is also a pure virtual class.
- Similar changes have been made to `LArProperties` and `DetectorClocks`
- Each experiment will have their own service and data provider that inherits from the pure virtual base class.

Status

- All code changes are in the `jpaley_LArPropertiesBreakup` feature branch. The *latest* version of LArSoft has not been merged into this branch, but it was merged a couple of weeks ago.
- Last night I completed all the changes to the `dunetpc` code to work with these changes. I then promptly accidentally deleted all my work on `dunetpc`.
 - The code changes are mostly straightforward, much of it can be done via `sed` commands
 - But there is a lot of code that contained unused `art::ServiceHandles`. I had done a fair amount of cleanup.
 - Can redo all this work in about a FT day (end of this week). Most of the time is spent waiting for `mrB` to actually start compiling the code.
- What else remains:
 - `fhicl` file changes
 - testing