

- Proposal: Remove `DataProperty` related functionality from `SupportFactory`; make it self-contained within `DataProperty`.
- Advocate: KTL

At the DC2 review, it was generally agreed that the `DataProperty` class should be a lightweight, self-contained manager of key/value pairs, suitable for use in many locations throughout the DMS code. In DC2, this goal was almost maintained, with one exception. The `SupportFactory` class, designed to annotate complex LSST data objects with metadata such as security and provenance information, was used to provide two utility functions related to `DataProperty`: `createLeafProperty()` and `createPropertyNode()`, of which the latter was the only mechanism for generating a `DataProperty` that could contain subsidiary `DataProperty`s. Accordingly, in conjunction with investigating the reconciliation of `DataProperty` with `Policy`, there seemed to be general agreement that the `SupportFactory` functionality related to `DataProperty` should be removed (and placed within `DataProperty` itself), while the class should remain for use with `LsstData` and `LsstBase` objects.

Making this change while the DC2 to DMS/DC3 transition is underway enables us to keep `SupportFactory` in the `daf/data` package where it belongs. Unfortunately, it also breaks a substantial amount of code that uses `SupportFactory` to create `DataProperty` root nodes. Such code would have to be rewritten to use a `DataProperty` method (a new constructor or a static method).

There is an alternative, made possible only because the `DataProperty` functionality is the **only** functionality currently located within `SupportFactory`. That alternate plan would be to move `SupportFactory` temporarily into the `daf/base` package, along with `DataProperty`, postponing the code updates until after the transition is complete.

Comment by robyn on Sun 30 Mar 2008 10:07:46 AM CDT

I agree that `DataProperty`? dependency on `SupportFactory`? should be removed.

With respect to the timing of implementing the full change:

- All files previously accessing `mwi/data` need to be reviewed to determine if the new version needs `daf/base` and/or `daf/data`.
- A single method is being moved from `mwi/data/SupportFactory` to `mwi/data/DataProperty`

Since each file is being examined anyway, I suggest the move of the single function from `SupportFactory`? to `DataProperty`? be done during the reorganization instead of temporarily moving the entire `SupportFactory`? class - which would necessitate a second later pass over the ~73 files affected during both moves.

Comment by RHL on Sun 30 Mar 2008 01:29:32 PM CDT

Do we still want to inherit from `Citizen`? My feeling is yes, as it enables leak detection. If we use `shared_ptr` properly this shouldn't be an issue, but it'd make me feel safer. We can of course gut `Citizen` (maybe with a compile-time switch analogous to `NDEBUG`) or remove it from `DataProperty` later.

Comment by JeffK (copied from E-mail by KTL) on Sun 30 Mar 2008 02:37:37 PM CDT

Just to provide some additional context, the original DC2 domain model explicitly EXCLUDED `DataProperty` from using `SupportFactory` in order to keep it more lightweight. This was always the intent.

While no longer part of the CCB, I strongly agree with this change.

Add comment

Your email or username: