

art - Support #20849

safety in art::DataViewImpl::getPointerByLabel(InputTag const& tag)

09/15/2018 07:28 PM - Pavel Murat

Status:	Closed	Start date:	09/15/2018
Priority:	Normal	Due date:	
Assignee:		% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Scope:	Internal	SSI Package:	
Experiment:	Mu2e		

Description

the function below is very handy.

It would be even more handy if, in case the corresponding product is not found, it returned a nullptr instead of unsuccessfully trying to dereference it.

-many thanks, Pasha

```
template <typename PROD>
inline PROD const*
art::DataViewImpl::getPointerByLabel(InputTag const& tag) const
{
    Handle<PROD> h;
    getByLabel(tag, h);
    return &(*h);
}
```

History

#1 - 09/17/2018 09:35 AM - Kyle Knoepfel

- Description updated

#2 - 09/17/2018 10:29 AM - Kyle Knoepfel

- Status changed from New to Feedback

We'd like to discuss with you some possibilities that do not require checking whether a pointer is null. We will contact you to set up a meeting.

#3 - 09/17/2018 10:30 AM - Kyle Knoepfel

- Occurs In 2.10.04 added

- Occurs In deleted (2.02.04)

#4 - 09/17/2018 11:47 AM - Kyle Knoepfel

Although getPointerByLabel is part of the public interface of Event (through art::DataViewImpl) we suspect exposing the bare pointer to the object is not the best design. We may decide to deprecate the function at some point, but let's first address the particular issue you brought up.

It sounds to me as if you would like to do something like:

```
auto p = e.getPointerByLabel<Product>(tag);
if (p == nullptr) {
    return; // or something similar
}
```

Something equivalent is already provided using the art::Handle system. Consider the following:

```
Handle<Product> h;
if (!e.getByLabel(tag, h)) {
    return;
}
```

```
// h is guaranteed to be valid if getByLabel returns true.  
assert(h.isValid());
```

The same number of lines are required as in the bare-pointer case--and no fragile dereferencing of a null pointer is possible. With C++17, the encapsulation is further improved:

```
// Using C++17  
if (Handle<Product> h; !e.getByLabel(tag, h)) {  
    return;  
}
```

Please let us know if the art::Handle solutions are still inadequate. We are happy to discuss further.

#5 - 11/07/2018 08:38 AM - Kyle Knoepfel

- *Tracker changed from Bug to Support*
- *Occurs In deleted (2.10.04)*

Did the art::Handle solution meet the need? If not, please let us know, otherwise we would like to close this issue.

#6 - 11/08/2018 02:46 PM - Kyle Knoepfel

- *Status changed from Feedback to Closed*
- *% Done changed from 0 to 100*

We are closing this issue due to discussion with Rob.