gallery - Feature #16547

Enable BranchDescription access through ProductID

05/15/2017 05:26 PM - Nathaniel Tagg

Status:	Closed	**Start date:**	05/15/2017
Priority:	Normal	**Due date:**	
Assignee:	Kyle Knoepfel	**% Done:**	100%
Category:		**Estimated time:**	1.00 hour
Target version:	1.10.00	**Spent time:**	2.00 hours
Scope:	Internal	**SSI Package:**	gallery
Experiment:	MicroBooNE		

Description

I would like to be able to figure out from the data file how associations connect objects.

The straightforward way for me to do that is to get an Assns<L,R> object, get the first item in the list, and look at the ProductID of each of the two lists it's associating. Then I want to figure out what label/process they each come from. That's trivially done by using productToBranch() in the BranchMapReader code, but that object is hidden as a private element of the DataGetterHelper which is in turn private inside the Event class.

Either making user-accessible get() calls for experts to dig down to the BranchMapReader, or providing top-level calls in Event would work.

I can trivially hack my own copy of the code base, but I'm planning to maintain downstream projects, and it would be convenient to have it patched into the release.

Here's how my code looks:

```
const gallery::BranchMapReader& bmr = ev.dataGetterHelper()->branchMapReader();
// these are the functions I hacked in to expose the internals

std::cout << "Associations " << std::endl;
typedef art::Assns<recob::Hit,recob::Wire> assn_t;
gallery::Handle< assn_t > assnhandle;

for(auto item: findByType<assn_t>(ev.getTTree())) {
  cout<< "LABEL " << item.second << std::endl;
  ev.getByLabel(item.second,assnhandle);
  if(assnhandle.isValid()) {
    std::cout << "Found " << assnhandle->size() << " associations." << std::endl;
    if(assnhandle->size()>0) {
      // Get first.
      std::pair<art::Ptr<recob::Hit>,art::Ptr<recob::Wire>> p = *assnhandle->begin();
      std::cout << p.first.id() << "\t" << p.first.key() << std::endl;

      const art::BranchDescription* desc1 = bmr.productToBranch(p.first.id());
      const art::BranchDescription* desc2 = bmr.productToBranch(p.second.id());
      std::cout << "Links " <<desc1->branchName() << " to " << desc2->branchName() << std::endl;
    }
  }
}
```

History

#1 - 05/17/2017 08:04 AM - Kyle Knoepfel

- Description updated

#2 - 05/22/2017 11:43 AM - Kyle Knoepfel

- Status changed from New to Feedback

An Assns product is designed to associate elements in potentially more than just two collections. In other words, there is no guarantee that the

ProductIDs of the second element in the Assns will match those of the first.

However, we would be able to provide a member function of gallery::Event that provides the BranchDescription of a product given its ProductID. Would this fit your need?

#3 - 05/22/2017 11:44 AM - Kyle Knoepfel

As a warning, due to ROOT IO needs, at some point it may become necessary to change the type that this member function returns.

#4 - 06/12/2017 12:02 PM - Kyle Knoepfel

- Status changed from Feedback to Accepted

Nathaniel wrote:

> True, in general an Assns can associate objects from more than two lists. In practice, however, this doesn't actually happen in any circumstance I'm aware of. This is a user problem, of course: the user would need to check each item to make sure they know.
>
> But yes, what I want is something to get a Branch Description (i.e enough to reconstruct it's InputTag) from the ProductID, which is what the code I show does.

#5 - 06/27/2017 11:48 AM - Gianluca Petrillo

I also second the request, as I have a unrelated but in the end not very different use case, where I assume a module which produced a product A to have also produced art::Assns<A,B>, and I am accessing A via an *art* pointer (so, no label, but product ID available).

#6 - 05/24/2018 02:10 PM - Nathaniel Tagg

This issue was put in a year ago and was 'accepted' but is not in the code. I have to maintain an entire parallel installation of gallery in order to keep my code alive. Can we please just put this in?

#7 - 05/30/2018 07:24 PM - Kyle Knoepfel

Hi Nathaniel, my apologies for this delay. So you're aware, we put issues in "Accepted" status whenever we agree to implement the proposed feature but are unable to commit up-front effort to it. Our time has been budgeted for implementing art 3.0, which will support concurrent processing of events. The multi-threaded *art* endeavor is close to being released (probably in the next week or so), at which point we'll be able to revisit our backlog of issues.

Again, my apologies for the delay. If you prefer it, feel free to adjust the priority level of this issue--part of the scheduling of new features involves taking into account their user-specified priorities.

#8 - 06/07/2018 02:43 PM - Kyle Knoepfel

- Subject changed from Expose productToBranch() to user to Enable BranchDescription access through ProductID

- Status changed from Accepted to Resolved

- Assignee set to Kyle Knoepfel

- % Done changed from 0 to 100

- SSI Package gallery added

This feature has been implemented with commits gallery:455671a6, gallery:dee12ff, and gallery:744691f. Although implementing it was fairly straightforward, some of the tests needed to be updated.

Example use:

```
auto const& pd = e.getProductDescription(productID); // => BranchDescription const&
std::cout << pd.inputTag() << '\n';
```

If the supplied product ID does not correspond to any product that is in memory, and exception will be thrown.

This feature will be included with the next minor version release of gallery.

#9 - 06/08/2018 10:32 AM - Kyle Knoepfel

- Target version set to 1.10.00

#10 - 06/14/2018 12:16 PM - Kyle Knoepfel

- Status changed from Resolved to Closed