

art - Bug #18738

PtrMaker can create pointers only for std::vector data products

01/12/2018 03:10 PM - Gianluca Petrillo

Status:	Closed	Start date:	01/12/2018
Priority:	Normal	Due date:	
Assignee:	Kyle Knoepfel	% Done:	100%
Category:	Application	Estimated time:	2.00 hours
Target version:	2.10.00	Spent time:	2.00 hours
Occurs In:	2.09.06	Experiment:	LArSoft
Scope:		SSI Package:	art

Description

The class art::PtrMaker (<source:art/Persistence/Common/PtrMaker.h#L55>) was written with the idea of supporting the creation of pointers from "any" container data products. Its relevant constructor:

```
//Creates a PtrMaker that creates Ptrs in to a collection of type C created by the module of type
MODULETYPE, where the collection has instance name "instance"
template <class EVENT, class MODULETYPE, class C = std::vector<T>>
PtrMaker(EVENT const & evt, MODULETYPE const& module, std::string const& instance = std::string
());
```

sports a data type C for the container data product itself.

But to my knowledge that template argument cannot be explicitly specified in a call, and it's not automatically detected either. That makes it impossible to effectively use the intended feature.

History

#1 - 01/16/2018 11:32 AM - Kyle Knoepfel

- Status changed from New to Accepted
- Estimated time set to 2.00 h

Oops.

#2 - 01/16/2018 11:32 AM - Kyle Knoepfel

- Description updated

#3 - 01/19/2018 02:55 PM - Kyle Knoepfel

- Category set to Application
- Status changed from Accepted to Resolved
- Assignee set to Kyle Knoepfel
- Target version set to 2.10.00
- % Done changed from 0 to 100
- Scope deleted (Internal)
- Occurs In 2.09.06 added
- SSI Package art added

#4 - 01/23/2018 12:03 PM - Kyle Knoepfel

- Status changed from Resolved to Closed

In order to facilitate the behavior you have requested, we have introduced the static member function 'create':

```
void MyProducer::produce(art::Event& e) override
{
    auto const maker = art::PtrMaker<int>::create<IntCollection>(e, *this, "my_instance");
    ...
}
```