



art product lookups and a bug

20 August 2019

art stakeholders meeting



art product lookups

- *art* always retrieves products based on a *product selector*, which accesses metadata exposed via the `art::BranchDescription` class
- Typically, users rely on a selector that only cares about:
 - C++ type of the product
 - Label of module that produced the product
 - Product instance name
 - Name of the process that produced the product (sometimes)
- Commonly used *art* product lookup facilities include

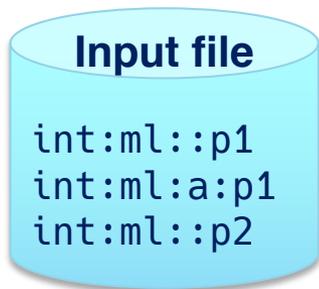
The interface below is ...	a specialized form of
<code>getValidHandle/getByLabel</code>	<code>getBySelector</code>
<code>getManyByType</code>	<code>getMany</code>

art product lookups

- A product can be retrieved without specifying the process name
- In this case, the lookup policy for `getValidHandle/getByLabel` is to
 - Iterate through process history in *reverse* order—i.e. start with current process, and then processes as stored on disk are consulted and traversed in reverse processing order
 - First match wins.
- Unfortunately, the definition of *first match* depends on the *art* version:
 - **art 3.02 and earlier**: the first match is the combination of locating the product **and** resolving it. Product lookup continues through the reverse process history until this combination is satisfied.
 - **art 3.03**: the first match corresponds to only locating the product, **not** resolving it. These changes were necessary to support a feature for DUNE.

The bug (<https://cdcvs.fnal.gov/redmine/issues/23054>)

- Consider this situation:



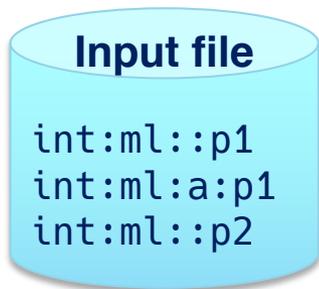
Current process (p3)

modules: **dl**, which looks up an int produced by ml in previous process,
ml, which creates int:ml::p3

TriggerPath: [dl, ml]

The bug (<https://cdcvs.fnal.gov/redmine/issues/23054>)

- Consider this situation:



Current process (p3)

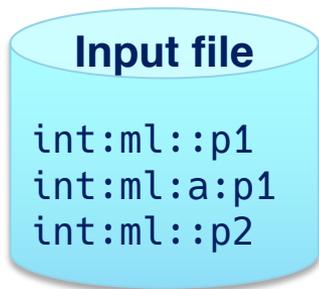
modules: **dl**, which looks up an int produced by ml in previous process,
ml, which creates int:ml::p3
TriggerPath: [dl, ml]

When processing dl:

```
e.getValidHandle<int>("ml");
```

The bug (<https://cdcvcs.fnal.gov/redmine/issues/23054>)

- Consider this situation:



Current process (p3)

modules: **dl**, which looks up an int produced by ml in previous process,
ml, which creates int:ml::p3
TriggerPath: [dl, ml]

When processing dl:

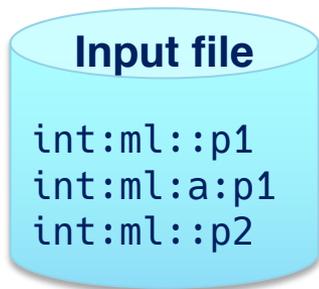
```
e.getValidHandle<int>("ml");
```

art 3.02:

Locate int:ml::p3

The bug (<https://cdcvns.fnal.gov/redmine/issues/23054>)

- Consider this situation:



Current process (p3)

modules: **dl**, which looks up an int produced by ml in previous process, **ml**, which creates int:ml::p3
TriggerPath: [dl, ml]

When processing dl:

```
e.getValidHandle<int>("ml");
```

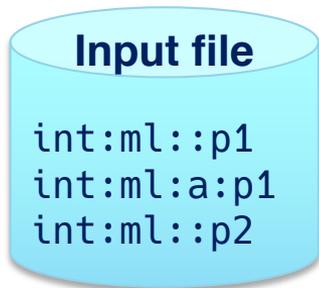
art 3.02:

Locate int:ml::p3

Product fetch failed

The bug (<https://cdcvcs.fnal.gov/redmine/issues/23054>)

- Consider this situation:



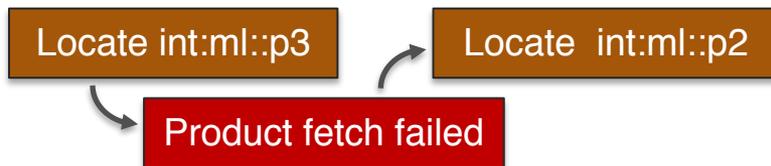
Current process (p3)

modules: **dl**, which looks up an int produced by ml in previous process, **ml**, which creates int:ml::p3
TriggerPath: [dl, ml]

When processing dl:

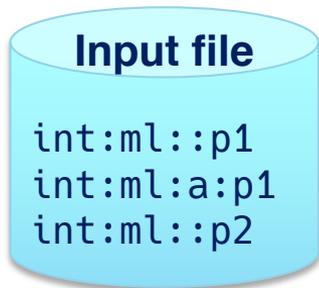
```
e.getValidHandle<int>("ml");
```

art 3.02:



The bug (<https://cdcv.s.fnal.gov/redmine/issues/23054>)

- Consider this situation:



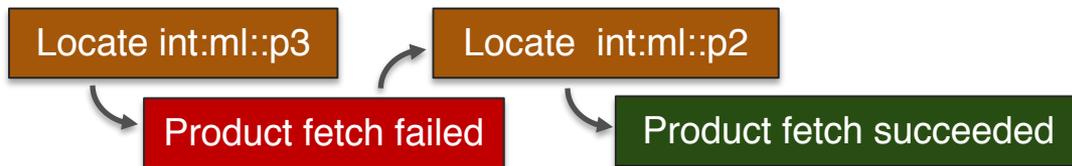
Current process (p3)

modules: **dl**, which looks up an int produced by ml in previous process, **ml**, which creates int:ml::p3
TriggerPath: [dl, ml]

When processing dl:

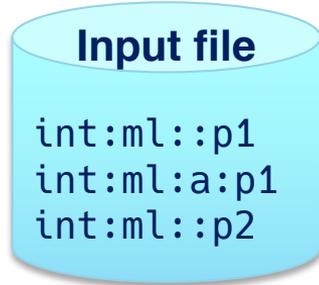
```
e.getValidHandle<int>("ml");
```

art 3.02:



The bug (<https://cdcv.s.fnal.gov/redmine/issues/23054>)

- Consider this situation:



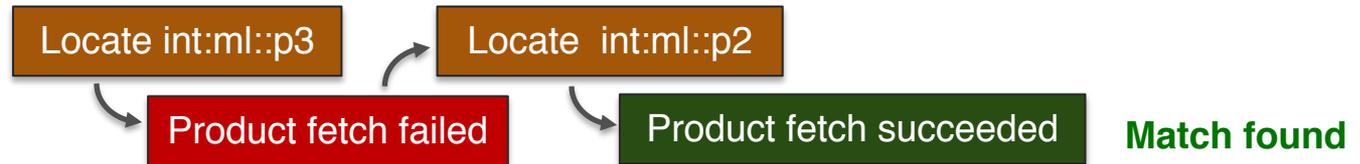
Current process (p3)

modules: **dl**, which looks up an int produced by ml in previous process, **ml**, which creates int:ml::p3
TriggerPath: [dl, ml]

When processing dl:

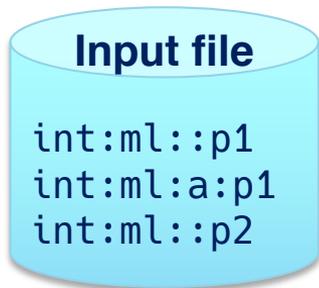
```
e.getValidHandle<int>("ml");
```

art 3.02:



The bug (<https://cdcvcs.fnal.gov/redmine/issues/23054>)

- Consider this situation:



Current process (p3)

modules: **dl**, which looks up an int produced by ml in previous process,
ml, which creates int:ml::p3
TriggerPath: [dl, ml]

When processing dl:

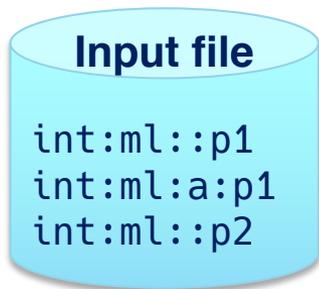
```
e.getValidHandle<int>("ml");
```

art 3.03:

Locate int:ml::p3

The bug (<https://cdcvs.fnal.gov/redmine/issues/23054>)

- Consider this situation:



Current process (p3)

modules: **dl**, which looks up an int produced by ml in previous process,
ml, which creates int:ml::p3
TriggerPath: [dl, ml]

When processing dl:

```
e.getValidHandle<int>("ml");
```

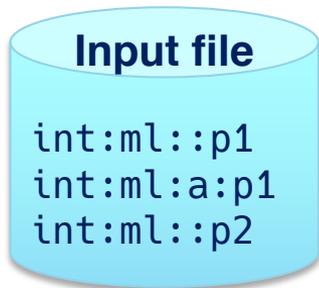
art 3.03:

Locate int:ml::p3

Match found

The bug (<https://cdcvns.fnal.gov/redmine/issues/23054>)

- Consider this situation:



Current process (p3)

modules: **dl**, which looks up an int produced by ml in previous process, **ml**, which creates int:ml::p3
TriggerPath: [dl, ml]

When processing dl:

```
e.getValidHandle<int>("ml");
```

art 3.03:

Locate int:ml::p3

Match found

Product fetch failed

Ways forward

- This is an *art* bug
 - A new version of *art* will need to be released on a timescale to be determined.
- You can avoid this problem by specifying the process name in the input tag (or the 'input_source' or 'current_process' special process names).
- My apologies.

Post-art 3.03

- Fixes for a few bug fixes to go in.
- Although there are features in demand, it is not yet determined how much work can be devoted to them.