

## art - Feature #6071

### Merging input files

04/29/2014 06:54 PM - Rob Kutschke

<b>Status:</b>	Closed	<b>Start date:</b>	04/29/2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Paul Russo	<b>% Done:</b>	100%
<b>Category:</b>	I/O	<b>Estimated time:</b>	120.00 hours
<b>Target version:</b>	1.14.00	<b>Spent time:</b>	80.00 hours
<b>Scope:</b>	Internal	<b>SSI Package:</b>	art
<b>Experiment:</b>	Mu2e		

#### Description

There has been some traffic on the artists mailing list regarding the issue of merging files. I would like to:

- Capture it here as an issue so that we can discuss it
- Explain what Mu2e wants
- **Explain a case that is likely too difficult to support - so don't paint yourself into a corner**

I have attached a pdf file with two figures.

Figure 1 shows an event processing workflow with 5 files. The solid arrows denote jobs that make file F2 from F1, etc. I will talk about the dashed arrows later.

F1 is a raw data file; it gets shipped to tape after use.

F2 is a reco file; it is the output of a job that used F1 as an input. F2 does NOT contain a copy of the raw data. It does have lots of new data products.

F3 is the output of a first skim step. F3 is the output of a job that used F2 as an input. It selects particular events; it adds some new data products and maybe drops a few. It's likely that one file F3 will contain events from several files F2.

F4 is the output of a second skim step. It selects particular events; it adds some new data products and maybe drops a few. It's likely that one file F4 will contain events from several files F3.

So F4 is the file that people will use for analysis. Probably most of the F2 and F3 generation of files will migrate to tape, leaving only the F4 family on disk. Eventually the analysis will identify some events of interest and we will want to use the event display to look, simultaneously, at raw data and data products from one or more of the files F2 through F4. This is represented by the box F5 and the dashed arrows.

**Now suppose that someone has managed to stage all of the F1...F4 files onto disk. We request that art be able to read all 4 files into a single file, F5.**

Why did I make a big deal about files migrating to tape? I am assuming that the experiment does not have enough disk space to keep everything they want on disk at one time. If you know in advance that you have enough disk space, then you can adopt a policy that F2 contains a copy of the raw data, F3 contains all of F2 plus new data products and so on. That is each generation only adds information and never drops it. In this case F4 will already contain the full processing history and this discussion is moot.

Figure 2 shows a different workflow. Again the solid arrows represent jobs that create one file from another.

F1 is a raw data file; it gets shipped to tape after use.

F2 is the output of a step we will call reco1; the job that writes F2 reads F1.

F3 is the output of a step we will call reco2; the job that writes F3 reads F1.

It is not important to say what reco1 and reco2 do; they could do something very different or just be two different versions of the reco code. We can do some analysis on the files F2 and F3. Based on this analysis we decide that we want to make an event display that looks at data products in all of F1, F2 and F3. This step is denoted by the box labelled F4 and the dashed arrows represent files to be read by this step.

The red slash indicates that art will not be able to support this workflow. The reason is the following: the art::ProductIDs used in F2 collide with those used in F3. An art::ProductID contains 2 integers, one that counts art jobs in the workflow and one that counts data products within an art job. In this example, F2 and F3 are both job number 2 within the workflow (neither knows about the other - each just knows that it follows the first job).

In the workflow shown in Figure 1, the processing chain is strictly sequential so it does not reuse art::ProductIDs.

In Figure 2, it will be possible to read data products from both F1 and F2 or F1 and F3, but not from any combination that includes both F2 and F3.

**We caution people not to design workflows like Figure 2 if they intend later to merge files.**

Another detail about Figure 1. For a single input file F4, there might be, say, 5 files F3, and 10 files each of F1 and F2. ( Why? because we presumed that the job that writes F3 might discard uninteresting events; we also assumed that we might produce one file F3 from several files F2; and we presumed the same about the job that writes F4 ). So art needs to be able to allow the each input stream to be composed of an arbitrary number of files.

In this ticket I have not addressed the issue of how someone would make sure that all of the necessary files are disk resident. Maybe dcache/samweb/xrootd are sufficiently powerful that we can just specify lists of input files to art and the staging will be automagic. Maybe not. If development is needed to make the staging work correctly, we believe that said functionality does not belong within art - it belongs in a separate tool.

**Related issues:**

Related to art - Bug #8897: Need restoration of erroneously-trimmed interface...	<b>Closed</b>	<b>05/22/2015</b>	<b>05/22/2015</b>
Has duplicate art - Feature #6557: multi-file product reading for ROOT input ...	<b>Rejected</b>	<b>07/01/2014</b>	<b>08/31/2014</b>

**History**

**#1 - 04/29/2014 07:04 PM - Christopher Backhouse**

Are there any additional requirements for ProductIDs to clash? eg, that the products are of the same type? or is it enough just for them to be from the same "generation"?

In any case, it sounds to me like we (NOvA) have only avoided having topologies like this (merging siblings) by blind luck.

Does there need to be a shared F1 to cause this problem? What if F2 and F3 both the first job in their workflow? I thought that was how our cosmic overlaying was done, so I'm not sure how comes that works.

**#2 - 04/29/2014 07:21 PM - Rob Kutschke**

One of the reasons I brought this up now was to call people's attention to the unsupported topology. It was only when I sketched the use case that we want that I started to explore variants and realized the problem.

I think that product types in several input files may be fully arbitrary so long as there are no ProductID clashes. In my Figure 1, I show merging from all 4 generations. There might be some product types that are the same and many others that are different.

I don't know what other constraints there may be - I will leave that for the art team.

You are right that if F2 and F3 are both the first job, then there will be clashes of ProductIDs.

One thing that I should have mentioned. I am talking about reading multiple input files via something like a modified RootInput source module. I am not talking about using the EventMixing mechanism. In event mixing, the mixed-in data products lose their provenance and their original ProductIDs; they are assigned new ProductIDs and their provenance shows that they came from a mixing module. If the mixed in data products contain embedded ProductIDs ( say in Ptr or Assns objects ), the user event mixing code is responsible for remapping them - the support tools for event mixing include a tool for remapping Ptrs. Does that answer your question about your cosmic overlays?

**#3 - 04/29/2014 07:40 PM - Christopher Backhouse**

Yes, we're using EventMixer for everything like this. So I guess we're safe from this.

**#4 - 04/30/2014 11:02 AM - Alden Fan**

Figure 1 covers exactly the use case that we (DarkSide) need.

**#5 - 05/05/2014 11:39 AM - Christopher Green**

- Category set to I/O
- Status changed from New to Accepted
- Target version set to 1.14.00
- Estimated time set to 120.00 h
- SSI Package art added
- SSI Package deleted ()

**#6 - 05/05/2014 11:39 AM - Christopher Green**

The time estimate is ballpark only.

**#7 - 01/29/2015 04:41 PM - Marc Paterno**

See

<https://web.fnal.gov/project/ArtDoc/news/Lists/Posts/Post.aspx?List=4aba71c2%2D90a3%2D4fb5%2D8add%2D7a37a4ec7947&ID=23&RootFolder=%2Fproject%2FArtDoc%2Fnews%2FLists%2FPosts&Source=https%3A%2F%2Fweb%2Efnal%2Egov%2Fproject%2FArtDoc%2FPages%2Fhome%2Easpx&Web=4b9a2ca7%2Dcb74%2D4510%2Dbe16%2D286fd7b5d9d8> for a description of the configuration options that have been discussed. We have agree to implement Option 1.

**#8 - 03/03/2015 09:54 AM - Christopher Green**

- Assignee set to Paul Russo

**#9 - 03/03/2015 11:14 AM - Christopher Green**

- Status changed from Accepted to Assigned

**#10 - 05/01/2015 09:14 PM - Christopher Green**

- Status changed from Assigned to Resolved

- % Done changed from 0 to 100

Implemented with merge [2a4e235f534b4e3aa9cc4123384afe7bfc8301bb](#).

Paul, please add your estimate of how much time you have spent on this feature. The time I have added is my estimate of my effort only.

**#11 - 05/04/2015 11:52 AM - Kyle Knoepfel**

- Status changed from Resolved to Closed

**#12 - 05/22/2015 11:44 AM - Christopher Green**

- Related to Bug #8897: Need restoration of erroneously-trimmed interface in art::OutputModule and art::ProductRegistryHelper added

## Files

---

FileMerging.pdf	21.8 KB	04/29/2014	Rob Kutschke
-----------------	---------	------------	--------------