art - Feature #5410

Allow MixFilter to set secondary filenames at runtime

02/10/2014 06:10 PM - Christopher Backhouse

Status:	Closed	**Start date:**	02/10/2014
Priority:	Normal	**Due date:**	
Assignee:	Christopher Green	**% Done:**	100%
Category:	I/O	**Estimated time:**	4.00 hours
Target version:	1.09.00	**Spent time:**	6.50 hours
Scope:	Internal	**SSI Package:**	art
Experiment:	-		

Description

Hi,

We have a job in NOvA where the secondary file to mix into the primary in a MixFilter job isn't available at runtime.

But there are no publicly available functions in MixHelper allowing the filenames_ and currentFilename_ variables to be set.

We'd like to request the addition of such a function. The initialization done in MixHelper::postRegistrationInit() also needs to be delayed until after the correct filename is set (or redone at that time).

Because the file is not available when writing the fcl script, a dummy filename needs to be tolerated, or a parameter added indicating that the filename will be set at runtime.

Thanks - Chris

Related issues:

Related to art - Bug #6346: registerSecondaryFileNameProvider and fileNames p...	**Closed**	**05/24/2014**
Related to art - Bug #6353: While mixing, the registered secondary filename p...	**Closed**	**05/27/2014**

Associated revisions

Revision 624fc5b7 - 02/25/2014 10:32 PM - Christopher Green

Implement issue #5410.

Detects respondToXXX detail functions and calls them at appropriate times.

New function MixHelper::registerSecondaryFileNameProvider().

Tests.

History

#1 - 02/20/2014 05:56 PM - Christopher Green

- Category set to I/O

- Status changed from New to Feedback

- Assignee set to Christopher Green

- Target version set to 1.09.00

- Estimated time set to 4.00 h

It looks like this should be do-able. Is the following interface acceptable: in the detail object's constructor, one should call the following function:

```
art::MixHelper::registerFileNameProvider(std::function<std::string ()> callback);
```

? An example would be (say):

```
helper.registerFileNameProvider(std::bind(&MyDetail::getSecondaryFileName, this));
```

I propose that in the first iteration the file should be a real file on disk, just like fileNames_ are expected to be now. I am not sure how much work would be involved in making it inter-operate with the SAM system. At the very least you would have to provide us with a URI or fully configure the interface.

Let me know if this is reasonable.

#2 - 02/20/2014 07:24 PM - Christopher Backhouse

How is the detail object intended to figure out what to return?

In my current implementation I have to inherit from MixFilter (instead of just specializing it with the details class) so that I can get respondToOpenInputFile(), which presumably isn't the intended usage.

Passing the FileBlock of the new primary to this callback would give me enough information. If not, I could stick with the implementation described above.

Interfacing with SAM would be slick, but it's not much trouble for my code to fetch the necessary file itself.

#3 - 02/25/2014 10:53 AM - Rob Kutschke

Christofer: what do you mean by "fetch" in your last reply. Are you proposing that your detail class knows how to move files around from one location (perhaps not on local disk) to local disk?

#4 - 02/25/2014 02:07 PM - Christopher Backhouse

Yes. I already have that part working.

But for the issue at hand, there's the same concern if I'm just looking up files on the local disk by eg applying a regexp to the input filename. You need to know the name of the primary input so that I can locate the secondaries, otherwise the options for what your detail class can do with this new ability are very limited.

#5 - 02/25/2014 02:40 PM - Rob Kutschke

It sounds like you and Chris will resolve this quickly enough but I am very concerned that you are trying to make the running art job all knowing and all seeing. I think that's a mistake because it eventually results in hard-to-maintain code.

For what it's worth, I think that a better answer is to have step in your grid script that resolves the correct filenames (which does mean interacting with SAM to get the first filename); then you can inject the correct filenames into the fcl you are done.

I will butt out of this ticket unless someone asks me back in.

#6 - 02/25/2014 10:39 PM - Christopher Green

- *Status changed from Feedback to Resolved*
- *% Done changed from 0 to 100*
- *SSI Package art added*
- *SSI Package deleted ()*

Implemented with 624fc5b.

MixFilter now detects and calls Detail::respondToXXXX functions.

MixHelper::registerSecondaryFileNameProvider(<func>) should be called from the constructor of your detail object to register a callback. This callback must be convertible to std::function<std::string ()>. A free function taking no arguments and returning std::string, a functor whose operator () has the same signature, or a bound free or member function whose signature after binding is std::string () are all convertible to std::function<std::string()>. See notes in MixHelper.h for details.

#7 - 02/26/2014 01:44 PM - Christopher Backhouse

Looks good.

I'll try it out once it's in a release that novasoft picks up (we're normally pretty quick about that) but I don't expect any trouble.

Thanks - Chris

#8 - 03/03/2014 11:45 AM - Lynn Garren

- *Status changed from Resolved to Closed*

#9 - 05/27/2014 02:13 PM - Christopher Backhouse

Unfortunately it looks like getSecondaryFilename() is called before respondToOpenInputFile(), which means I can't get hold of the necessary information in time.

Is it easy to swap the order those calls wind up occurring?

#10 - 05/27/2014 03:16 PM - Christopher Green

Since this is a closed issue, I have entered a new issue #6353, and am investigating now.