

# Sam and Luminosity

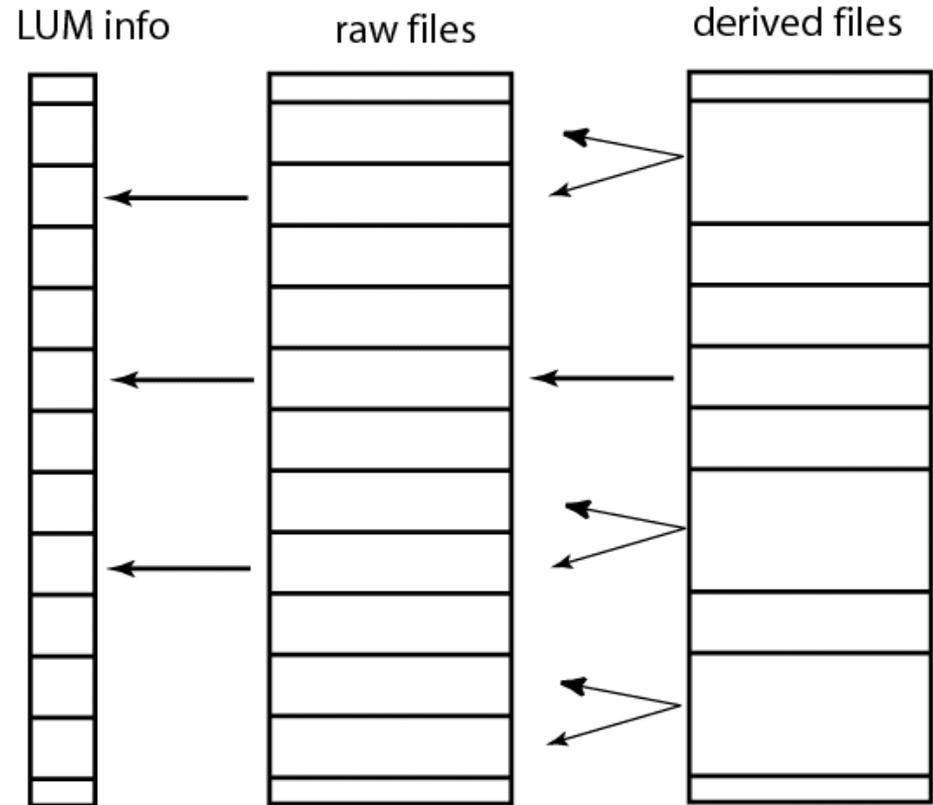
H. Schellman  
October 24, 2002

Sam tries to track what you do so you  
can normalize later

- Sam stores
  - your definition
  - what files were actually delivered
  - what processes were run on those files
  - what processes made those files
- In principle, you can figure out everything done to a file since it was logged online

# Luminosity information is associated with files

- At D0, the luminosity for your analysis is found by associating the files in your analysis with luminosity information.
- This makes knowing which files were in your analysis vital.
- This information is stored in the file metadata



## What we know about derived file

recoT\_all\_0000164605\_mrg\_210-213.raw\_p11.12.01

```
runNumber: 164605
physicalDatastreamName: all,
dataTier: thumbnail,
eventCount: 8852
lumMin: 1585395, lumMax: 1585398,
version: p11.12.01, applName: recon_root,
projectName: farm.p11.12.01.18157,
projSnapId: 38056,
projectDefName: farm-dayset-2002-09-24-164605-2-
p11.12.01_20020925163504
children list: []
parent list: ['all_0000164605_210.raw',
'all_0000164605_211.raw', 'all_0000164605_212.raw',
'all_0000164605_213.raw']
```

```
sam dump file --filename=<name>
sam get metadata --filename=<name>
```

# Analysis examples

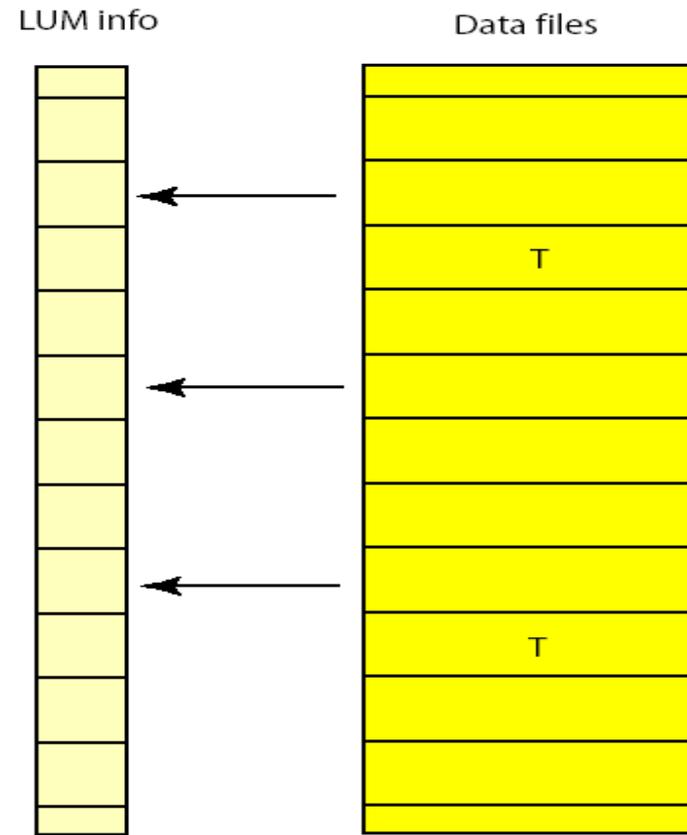
- Analysis actually consists of two steps
  - Produce your analysis sample
  - Run the analysis job over it many times.
- If you do these right, you can get accurate luminosities
- Example - analyzing a very rare trigger without exclusive streaming

## First step

- Your physics group probably wants to create a derived sample by filtering based on triggers/event cuts.
- Set up a production run, run over all of the data. Do all the book-keeping right
- You may also wish a sample of events for quick and dirty work - pick events. It is hard to get accurate luminosity for pick events samples.

# Correct example

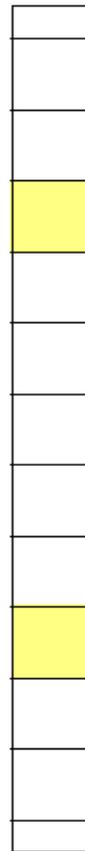
- Your data actually only has trigger T in 2 files but the list of files that trigger T could have been in is much larger.
- The luminosity corresponds to all of the time that trigger T was live, not just when it fired.
- Your correct list of input files is the full list, not just the ones with trigger T in them.
- If you do filtering, and just store T events, you need to include all of the other files in the parentage because T could have been in them.



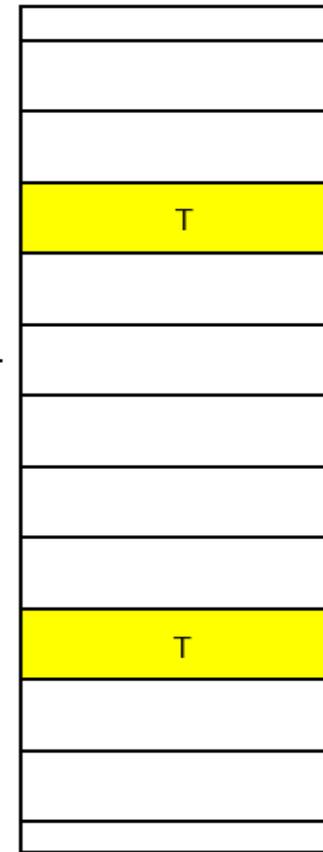
# Bad example

- You logically say, why get all the files, I only need the ones with my trigger in them...
- Unless you get that set of files carefully, you will only get a small fraction of the luminosity.
- Pick events is this example!

LUM info

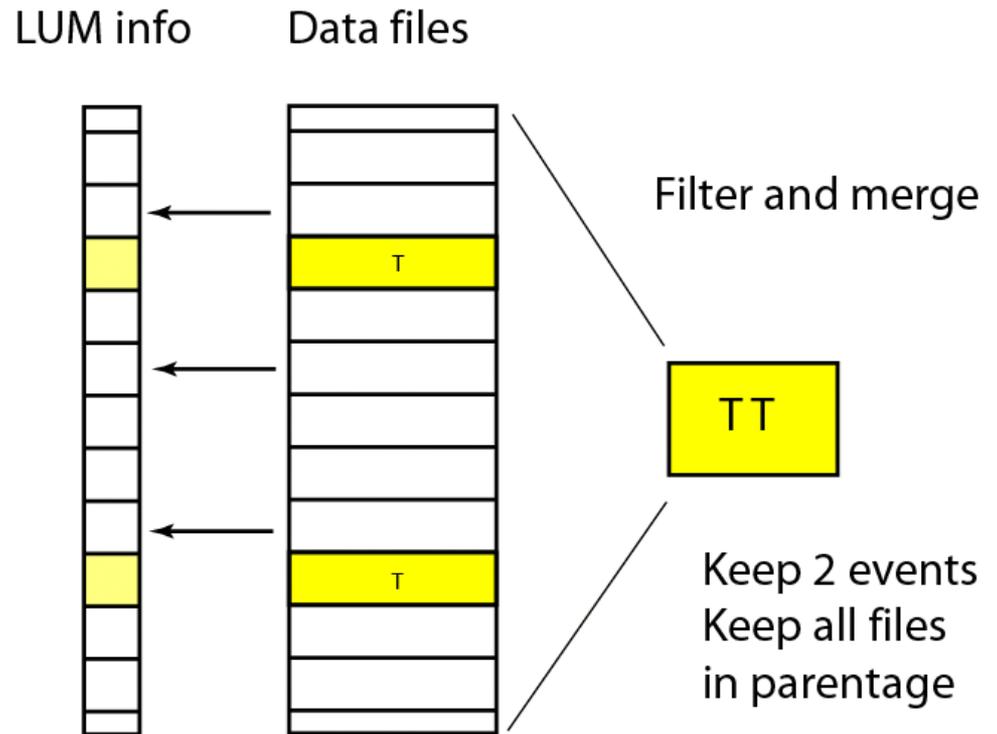


Data files



# Solutions

- Production filtering merges files and includes all input files in the parentage, your big set of files becomes one smaller file which has pointers back to all of the luminosity.



## Another method...

- Make 2 sam definitions, one with all data that trigger was live for, the other for those files which actually have events. Use list of files in the first one to derive your luminosity.
  - Quality cuts?
  - Reconstruction losses?
  - Bad runs you forgot about?

# Lm\_access

- Lm\_access is the luminosity package.
- As you spin a data sample, it uses the filenames to make a list of lbn's for that sample, cuts out the bad ones, and sums up the good ones.
- It depends on two DB tables
  - A parentage DB, which maps filenames for your input onto raw datafiles -> lbn's
  - The luminosity DB keyed on lbn number.
- If the parentage is messed up this does not work.

# Can I normalize this?

- Unless all of the input files corresponding to the trigger list or run range for your analysis are included in the parentage, these derived files don't have the information in them to get a luminosity directly.
- In principle you use sam dataset definitions to get the list of parent files you should have had and use that for normalization. But this is not protected against errors.
- Production files do have the full information in their metadata.

# Skimming carefully

- Write out one event/job no matter what - then those files are in parentage as you did look at them.
  - `rcp <special_stream SpecialStream>` in `framework.rcp`
- Close output files on input boundaries
  - `bool Synchronize = true` in `WriteEvent.rcp`
  - `string outputfile = "SAMGenerated:"` is a nice way to generate unique filenames
  - `int InputFilesPerFile=N` allows you to control the output file size.
- Count parentage when input file closes, not opens
  - `int FileParentageMode = 1` in `sam_manager.rcp`
- Check that file with same processing and parentage is not already in sam before storing.
  - On the way....

# Do I really have to do this?

- Thanks to Herb Greenlee and Marco Verzocchi and the sam/tools teams we are close to having utilities that do event skimming I/O for you. You still have to decide on what you want.

# Storing files back into sam

If you use sam for input, and write output in DST or thumbnail format, you will get an output file:

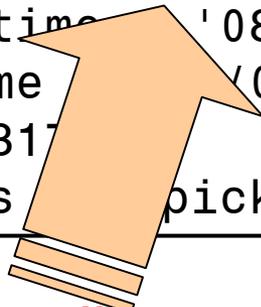
`<outputfile>.metadata.py`

as well.

With a little editing you can use this to store your output back into sam.

This is metadata produced by a copysam.py job which merged two input files which came from sam, pick\_w32.dat and pick\_w1.dat

```
from import_classes import *
TheFile = ProcessedFile(
    name = '2files',
    sizeK = 40104,
    events = Events(3654425, 641460, 198),
    stream = '',
    tier = 'reconstructed',
    start_time = '08/02/2002 19:56:00',
    end_time = '08/02/2002 19:56:05',
    pid = 817,
    parents = ['pick_w32.dat', 'pick_w1.dat'])
```



Currently sam calls everything reconstructed - you have to change this before you store the file! You must use a <data\_tier-bygroup>

# What data-tier to use in storing

- Data-tier tells what kind of data it is
- Raw, sim, reconstructed, thumbnail, root-tuple...
- There are two qualifiers:
  - filtered- - indicates that not all of the parent events were passed through
  - bygroup -- means it's a sample produced by a user or physics group rather than general production.
  - Tom Diehl stores pick events WZ samples as raw-bygroup. Probably should be filtered-raw-bygroup...

# What's that stream variable

- It is not the physical stream written online, it's a special tag for you to use to tell derived datasets apart.
- Right now there seems to be no way to use it though, still should fill it in so you can access it later.
- MC group has come up with a whole parameters schema which may also be very useful for these derived sets.

# To store data you need

- A **file** to store - with a unique name - please don't use something that looks real official - others may pick up your file in a query...
- Valid **metadata** for that file, generated by the framework if you use sam input and EVPACK output. (and then edited to have the right data\_tier - yours must be of form x-bygroup.
- A **pnfs** location to store it to - ask your physics/id group boss.
- The WZ group has a script which does this ...

```
sam store --descrip=<meta.py> --source=$PWD  
--dest=<your pnfs location>
```

# How data stores work

Sam has a file storage server (fss) which takes your file and metadata.

It puts the metadata into sam, then uses enstore to copy the file into the tape robot.

Enstore maps directories to sets of tapes. Your group needs to have a directory set up in order to store files.

```
/pnfs/sam/dzero/copy1/physics_data_taking/group_phase1/top/thumbnail/all
```

Once the store to tape is done, the location (tape volume) is added to the metadata.

You either need to know the full destination path or use auto-destination, which keys on group, data\_tier, stream and finds the right location.

# Checking on file stores

On the machine you are storing from (usually d0mino)

```
sam dump fss
```

```
ps -ef | grep eworker | grep <file>
```

Or look at:

[http://www-d0en.fnal.gov/enstore/enstore\\_system.html](http://www-d0en.fnal.gov/enstore/enstore_system.html)

[http://www-d0en.fnal.gov/enstore/status\\_enstore\\_system.html](http://www-d0en.fnal.gov/enstore/status_enstore_system.html)

[http://www-d0en.fnal.gov/enstore/enstore\\_files.html](http://www-d0en.fnal.gov/enstore/enstore_files.html)

# Merging small output files

- If your output files are small, you need to merge before you store as tapes are big.
- Farms have scripts which can merge files before they go into sam but they depend on parsing the filename for some important information - not general
- WZ group has the ability to merge files which are already in sam to make bigger more convenient ones.
- But you really want to merge the files before you store them. Preliminary script from Marco, to do it with full checking for overlaps, need some more sam coding.

# Conclusion

- Making derived samples takes some care if you want precise luminosities for those samples.
- Production does this for you
- Physics groups should get together and do careful skims for their samples.
- Tools are almost there to do this. Your input on needs, use cases would be very welcome.