Notes on SAM

SAM is (mainly) a data catalog originally designed for the D0 and CDF high energy physics experiments at Fermilab.

The most important objects cataloged in sam are individual files and collections of files called **datasets.**

Data files themselves are not stored in sam, their metadata is and that metadata allows you to search for and find the actual physical files.

Sam was designed to ensure that large scale data-processing was done completely and accurately which leads to some features not always present in a generic catalog but very desirable if one wishes high standards of reproducibility and documentation in data analysis.

1st) The metadata is associated with the file name itself. This makes renaming a file very unwise. This also means that your filenames must be unique. A very common practice is to include some of the metadata in the filename, both to make it easier to identify and to ensure uniqueness. One obvious thing to do, if running private processing, is to put your username in the filename to own it.

2nd) That metadata can include file locations. A file can have no location at all, or many. When you move or remove a file with an associated sam location, you need to update the location information.

3rd) sam has associated tools that can actually move physical files between locations, either storing them or making temporary copies. Or it can just tell you where the file is and let you use that information to copy the file using your own methods. Storing and removing files requires both moving/removing the file itself and updating its metadata to reflect that location and is generally left up to special packages such as FTS.

4th) Files which are stored on disk or tape are expected to have appropriate file sizes and checksums. One can have duplicate instances of a file in different locations but they must all be identical. If one reprocesses a file, it may well be subtly different (for example dates stored in the file itself can change the checksum). Sam should force you to choose which version, old or new, is acceptable. It will not let you catalog both with the same filename. As a result, if you get a named file out of sam, you can be reasonably certain you got the right copy. As we have recently learned, not all software companies are as careful as sam about the integrity of their copies.

5th) files with duplicate content but different names can be problematic. The reprocessed file mentioned in part 4, if renamed, could cause significant problems if it were allowed in to the data sample as a job processing all files might get both copies. This is one of the major reasons for the checksums and unique filenames. There is a temptation to put, for example, timestamps, in filenames to generate unique names but that removes the protection against duplication.

6th) files can have parents and children and, effectively birth certificates that can tell you how they were made. An example would be a set of raw data files RAWnnn processed with code X to produce a single reconstructed file RECO. One can tell sam that RAWnnn are the parents of RECO processed with version x of code X. If one later finds another RAWnnn file that was missed in processing, sam can tell you it has not been processed yet with X (i.e., it has no children associated with version x of X) and you can then choose to process that file.

The D0 experiment required that all official processing going into sam be done with tagged releases and fixed parameter sets to increase reproducibility. Constants databases were harder to timestamp so some variability was still possible if calibrations were updated.

7th) metadata can also contain "spill" or luminosity block information that allows one to point to specific data taking periods with smaller granularity than a run or subrun. When files are merged, this spill information is also merged.

All of these features are intended to assure that your data are well described and can be found. As sam stores full location information, this means any sam visible location. In addition, if parentage information is provided, you can determine and reproduce the full provenance of any file.

In addition to the files themselves, sam allows you to define datasets.

A dataset is effectively a query against the sam database. An example would be "data_tier reconstructed and run_number 2001 and version v10" which would be all files from run 2001 that are reconstructed data produced by version v10. This dataset is dynamic. If one finds a missing file from run 2001 and reconstructs it with v10, the dataset will grow. There are also dataset snapshots that are derived from datasets and capture the exact files in the dataset when the snapshot was made.

Suggestions for configuring sam.

First of all, it really is nice to have filenames and dataset name that tell you what's in the box, although not required. The D0 and MINERvA conventions have been to use "_" underscores between useful key strings. As a result, D0 and MINERvA tried not to use "_" in metadata entries to allow cleaner parsing. "-" is used if needed in the metadata.

D0 also appended processing information to filenames as they moved through the system to assure that files run through different sequences had unique identifiers.

Example: a Monte Carlo simulation file generated with version v3 and then reconstructed with v5 might look like

SIM_MC_020000_0000_simv3.root would be a parent of

RECO_MC_020000_0000_simv3_recov5.root

Data files are all children of the raw data while simulation files sometimes have more

complicated ancestry, with both unique generated events and overlay events from data as parents.

Setting up sam metadata:

This needs to be done once, and very carefully, early in the experiment. It can grow but thinking hard at the beginning saves a lot of pain later.

You need to define **data_tiers**. These represent the different types of data that you produced through your processing chain. Examples would be raw, pedsup, calibrated, reconstructed, thumbnail, mc-generatod, mc-geant, mc-overlaid,

run_type can be used to tell test beam from near detector from …

data_stream is often used for trigger subsamples that you may wish to split data into (for example pedestal vs data runs).

Generally you want to store data from a given data_tier with other data from that tier to facilitate fast sequential access. This gets us into file_families which are a pnfs concept that writes similar data to tape.

Applications

It is useful, but not required to also define **applications** which are triads of “appfamily”, “appname” and “version”. Those are used to figure out what changed X to Y. There are also places to store the machine the application ran on and the start and end time for the job.

```
samweb list-files "data_tier raw and not isparentof: (data_tier reconstruc
ted and appname reco and version 7)"
```

Should, in principle, list raw data files not yet processed by version 7 of reco to produce files of tier reconstructed. You would use this to find lost files in your reconstruction after a power outage.

Merging and splitting

Parentage works pretty well if one is merging files but splitting them can become problematic - partially as, once merged, the spill information is hard to split back.

Sam will let you merge files with different attributes if you don’t check carefully. Generally it is a good idea not to merge files from different data tiers and certainly not from different data ty pes. Merging across major processing versions is probably also unwise.