

Production Operations – Version 1.1

Last updated: 4/29/15

by: Katherine Lato

With contributions by:

Paola Buitrago, Anna Mazzacane, Andrew Norman, Jeny Teheran Sierra, and Marek Zielinski.

Abstract

Production Operations takes the raw data from experiments and transforms it through a series of stages to a format that allows physicists to do analysis on the data.

Revision History:

Version 1.0 – 3/19/15	Base document
Version 1.1 – 4/29/15	Adding NOvA Detailed Example

Table of Contents

1 Production Operations Background.....	4
1.1 Overview.....	4
1.2 KeepUp (Raw2Root) Stage.....	5
1.3 Calibration Stage.....	5
1.4 Reconstruction Stage.....	5
1.5 Particle Identification Stage.....	5
1.6 Analysis Stage.....	5
2 NOvA Production Operations	6
2.1 NOvA Overview.....	6
2.2 NOvA Detailed Example.....	7
2.2.1 Data Tiers	7
2.2.2 Naming Datasets.....	7
2.2.3 Raw2Root KeepUp.....	8
SAM Dataset dimensions.....	8
Raw2Root KeepUp submission.....	8
Arguments.....	8
Raw2Root KeepUp as cron jobs.....	9
Notes.....	9
2.2.4 Raw2Root Backprocessing.....	9
2.2.5 Reconstruction Keepup.....	9
SAM Dataset dimensions.....	10
Scripts.....	10
Reconstruction as cron jobs.....	11
Notes.....	11
2.2.6 Reconstruction Backprocessing.....	11
2.3 NOvA Electronic Log Book.....	12
2.4 Raw2Root for NOvA	12
2.4.1 Running the cron job.....	13
2.4.2 Raw2Root Pattern Dataset Dimensions.....	13
2.5 PChits.....	14
2.5.1 Running the cron job.....	14
2.5.2 Naming format.....	14
2.5.3 PChits pattern.....	15
2.6 NOvA Monitoring.....	15
2.7 Special Situation - Run raw2root manually for NOvA	16
2.8 Condor logs archival for novapro	17
2.9 novapro Credentials.....	17
3 MINOS.....	18
3.1 Script Overview.....	18
3.2 KEYGEAR Script.....	18
3.3 AMBROSIA Script.....	21
3.4 Useful Information.....	22

3.5 Minos Monitoring.....	22
4 MINERvA.....	22
4.1 Monte Carlo Production.....	22
4.2 Data Production.....	23
4.3 User Analysis.....	23
5 Production Tools.....	23
6 Glossary.....	24
7 Appendix A.....	25

1 Production Operations Background

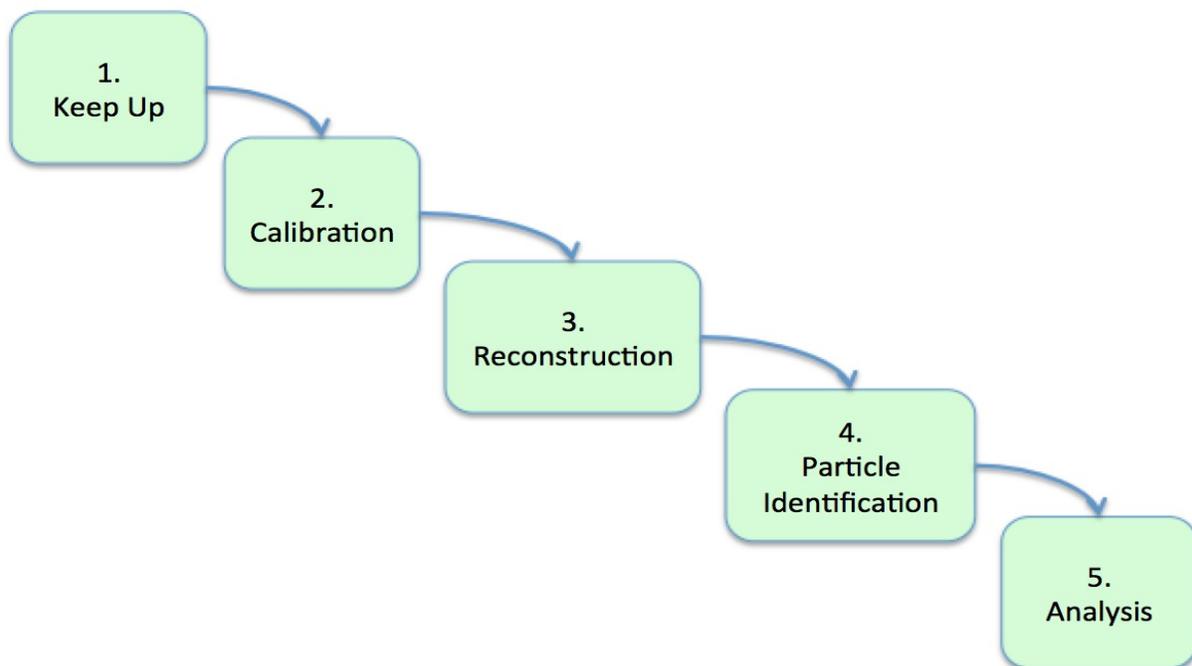
1.1 Overview

Staff submits, monitors, and validates the offline work flows and analysis suites (simulation, calibration, reconstruction) for Fermilab experiments performed on computing resources both at the Fermilab Site and on the Open Science Grid.

Current projects are: Minos, Minerva, Nova

Current staff are: Paola Buitrago, Anna Mazzacane (end of January 2015), Jeny Teheran Sierra, and Marek Zielinski.

At a high-level, can think about offline production as consisting of five stages:



The number of files handled daily depends on the experiment. NovA has over 5,000 raw input files per day. Other experiments process fewer files. The raw data is in a bit pattern that isn't useful to humans. There are a number of stages needed to get from the raw input files to the stage that allows physicists to do analysis on the data.

The goal is to automate as much of the translation, calibration and reconstruction as possible with scripts and to monitor to ensure the scripts worked correctly. The experiment gives:

- data
- scripts to run

The staff:

- runs jobs and looks at the results
- catalogs the results
- monitors the workflow
- Looks for improvements to make.

1.2 KeepUp (Raw2Root) Stage

The Raw2Root process translates the raw files into files in the ROOT Data Analysis framework. (For more information on Root, see:

<http://root.cern.ch/root/html/doc/guides/primer/ROOTPrimer.html#welcome-to-root>)

1.3 Calibration Stage

Calibration and renormalization of data is done to have it make sense. This stage converts raw electronics into something of interest in physics, energy. The result is another ROOT file, but it's calibrated.

1.4 Reconstruction Stage

The reconstruction stage is connecting the dots to get the real trajectory of particles. This is understanding the topology. Complex algorithms are run with statistical groupings. The output is more physics-like, tracks.

1.5 Particle Identification Stage

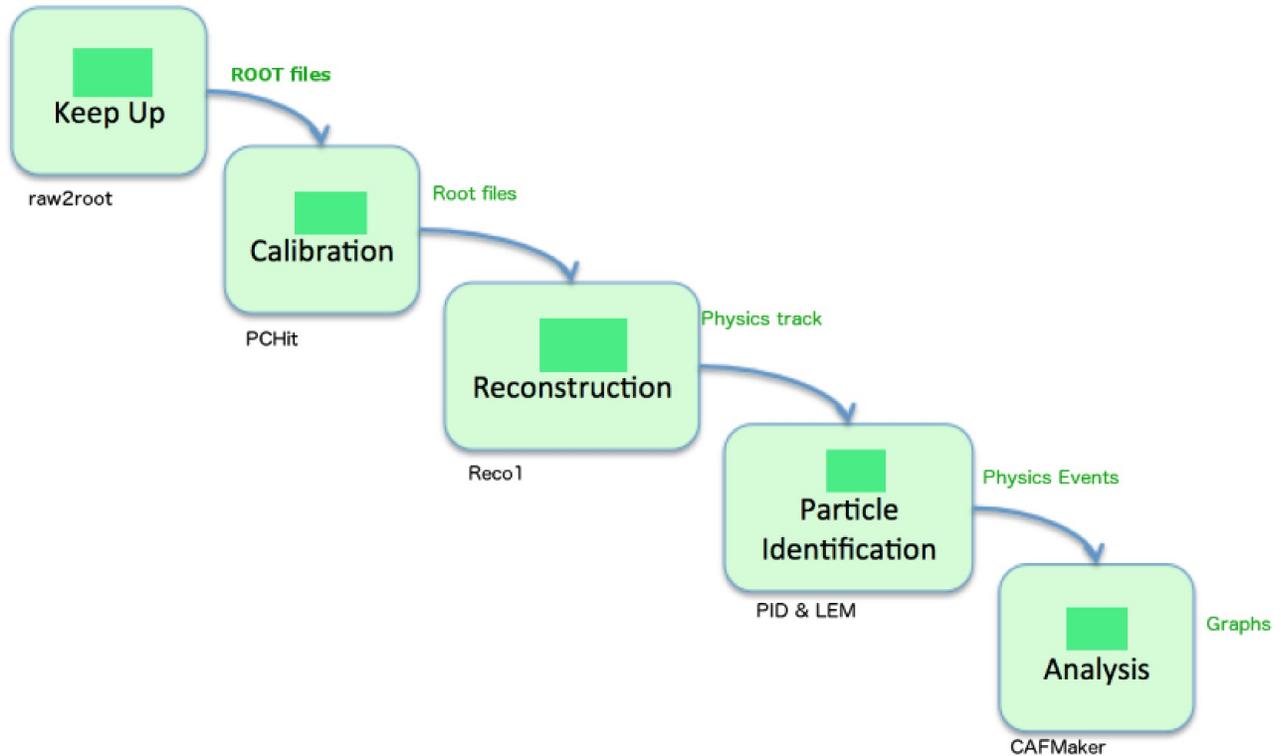
This stage involves classifying the objects. The output is fully understood events.

1.6 Analysis Stage

Physicists perform the analysis stage. They count different topologies, determining how much energy is used. The output is a histogram.

2 NOvA Production Operations

2.1 NOvA Overview



1. Detector records raw files.
2. Raw files are processed to make artdaq files.
3. Artdaq files are processed to make pclist, pcliststop & timecal files.
4. Pclist, pcliststop & timecal files are used to make calibration constants which are stored as csv files or in a database.
5. raw files and calibration constants are used to run reconstruction to produce reco files.
6. recon files and calibration constants are used to run particle ID and produce pidpart files and LEMpart files.
7. LEMpartfiles are used to make LEM files. (This stage is, uniquely, run on dedicate resources as caltech.)
8. pidpart files and LEM files are combined to make PID files.
9. PID files are used to make CAF files.

Note:

*6,7,8 can run as a single job (but LEM requires hundreds of gigs of memory so is typically forked into

a separate process on a dedicated farm).

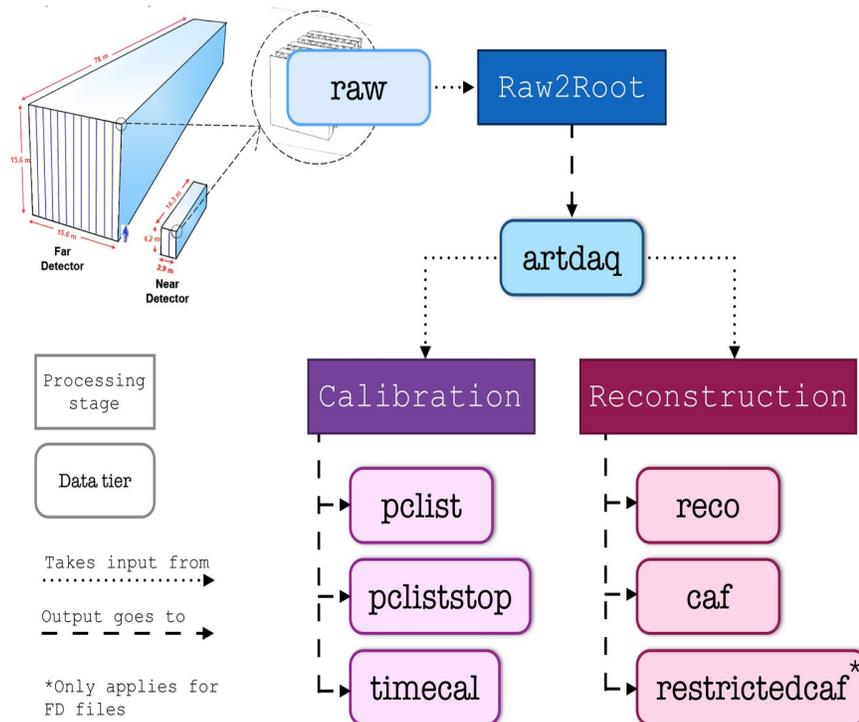
* 5 & 6 don't need calibration constants to run

* 9 can run off of reco files as well as PID files

* 3 actually uses calibration constants itself (albeit older versions), so in a sense our calibration is iterative.

2.2 NOvA Detailed Example

2.2.1 Data Tiers



2.2.2 Naming Datasets

For all the processing stages of NOvA Offline Production (Keepup), there are input and output datasets defined by SAM dimensions. In order to follow a standard, we suggest using this pattern for input or output datasets:

```
prod_[data_tier]_[nova_release]_[detector_id]_[stream]_keepup.[additional_data]
```

For example:

- Raw2Root input dataset: prod_raw_FA14-10-03_nd_others_keepup.2015-04-26
- Reconstruction input dataset: prod_artdaq_S15-02-05_nd_numi_keepup.2015-04-18

- Reconstruction output dataset: prod_reco_S15-02-05_nd_numi_keepup.r10496_r10798

2.2.3 Raw2Root KeepUp

Raw2Root is the first stage for NOvA Offline Production. It will take raw data to produce ROOT files. The output files for this stage will belong to *artdaq* tier.

SAM Dataset dimensions

Dimensions for datasets (draining dataset) that will be used as input of this stage:

```
file_type      = 'importedDetector' and
data_tier      = 'raw' and not
isparentof:   ( data_tier = 'artdaq' ) and
Online.Detector = '${DET}' and
Online.SubRunEndTime >= '${START_DAY_SECONDS}' and
Online.SubRunEndTime <= '${END_DAY_SECONDS}' and
Online.Stream = $STREAM and
Online.TotalEvents > '0'
```

Raw2Root KeepUp submission

Raw2Root has two steps: input dataset generation and job submission. Raw files are collected every day in the NOvA Detectors, Raw2Root needs to look for the files that were created in the last previous days and process them. The script @Raw2RootKeepUp.sh@ will look for the files created up to 1 month ago iterating over odd dates. A maximum of 16 datasets will be created per Raw2Root running.

```
*Raw2RootKeepUp.sh*
```

Usage:

```
$ Raw2RootKeepUp.sh <arguments>
```

Arguments

The name of the detector which the files belong to must be specified, along with the name of the stream and the NOvA release to use to process those files.

```
--det DETECTORNAME   Detector name: fardet or neardet
--stream STREAM       SAM dataset definition to run over
--release NOVARELEASE NOvA Release under Raw2Root will process files
```

Raw2Root KeepUp as cron jobs

Raw2Root Keepup involves a set of cron jobs running daily. A single cron job will process one detector and one stream: *NuMI*, *Cosmics* or *Others*. It will start with Far Detector files, processing *NuMI*, *Cosmics* and *Others* streams separately with @S14-08-19@ release. It will continue with Near Detector files, following the same order for streams with @FA14-10-03@ release. There are 6 cron jobs placed in a crontab to execute Raw2Root KeepUp.

Raw2Root processing will start at 01:00 a.m.

Crontab list output (novagpvm01.fnal.gov)

```
00 01 * * * /nova/app/home/novapro/KeepUp_opos/Raw2Root_KeepUp/Raw2RootKeepUp.sh --det fardet
--stream numi --release S14-08-19
00 03 * * * /nova/app/home/novapro/KeepUp_opos/Raw2Root_KeepUp/Raw2RootKeepUp.sh --det fardet
--stream cosmics --release S14-08-19
00 05 * * * /nova/app/home/novapro/KeepUp_opos/Raw2Root_KeepUp/Raw2RootKeepUp.sh --det fardet
--stream others --release S14-08-19
00 07 * * * /nova/app/home/novapro/KeepUp_opos/Raw2Root_KeepUp/Raw2RootKeepUp.sh --det neardet
--stream numi --release FA14-10-03
00 09 * * * /nova/app/home/novapro/KeepUp_opos/Raw2Root_KeepUp/Raw2RootKeepUp.sh --det neardet
--stream cosmics --release FA14-10-03
00 11 * * * /nova/app/home/novapro/KeepUp_opos/Raw2Root_KeepUp/Raw2RootKeepUp.sh --det neardet
--stream others --release FA14-10-03
```

Notes

- Since November 2014, Near Detector Raw2Root KeepUp Processing is done under FA14-10-03. Due to an error in geometry detector, files from run number = *10377* up to *10620* were reprocessed. A set of raw files have two children files under different releases.
- Since December 12th, 2014, new dimensions for Raw2Root input datasets were defined. After an error reported for NOvA production, dimensions were fixed.
- Since April, 2015, new dimensions for Raw2Root input datasets were defined. After a request from NOvA production, @SubRunStartTime@ metadata field was replaced by @SubRunEndTime@.

2.2.4 Raw2Root Backprocessing

When a Raw2Root Backprocessing request comes from NOvA experiment, the input dataset is created using the NOvA Editor.

2.2.5 Reconstruction Keepup

Reconstruction stage takes files from the *artdaq* tier and will produce output files in *reco*, *caf*

tiers.

SAM Dataset dimensions

Dimensions for datasets (draining dataset) that will be used as input of this stage:

```

file_type      = 'importedDetector' and
data_tier      = 'artdaq' and
Online.Detector = '${DET}' and
Online.Partition = 1 and
Online.SubRunStartTime >= '${START_DAY_SECONDS}' and
Online.SubRunStartTime <= '${END_DAY_SECONDS}' and
Online.Stream = $STREAM and not
isparentof:   ( data_tier = 'reco')

```

Scripts

Reconstruction scripts are located in: /nova/app/home/novapro/KeepUp_oog/Reconstruction_KeepUp

cron_KeepUp.sh

This script has those input parameters: Detector name: {fardet | neardet}, Stream: {0 | 2 | -1 }, Release

This script will export variables and execute the dataset generation script for Reconstruction KeepUp.

ReconstructionKeepUp.sh

This script has those input parameters: Detector name: {fardet | neardet}, Stream: {0 | 2 | -1 }, Release

This script will setup the environment and build the datasets using the dimensions defined before.

Once the datasets are ready, this script will execute the submission script ***submitJob_keepup.sh***.

submitJob_fd_keepup.sh* | *submitJob_nd_keepup.sh

This script has those input parameters: Detector name: {fardet | neardet}, Dataset name, Days ago, Release

There are different submission scripts for both detectors, because the fcl files are different and the jobsub command includes additional output data tiers.

Once the reconstruction cron jobs are stable, an unified submission script will be built

This script will submit the jobs to the grid using ***jobsubs_tools***.

Reconstruction as cron jobs

Reconstruction is setup as a daily cron job. It will process Far and Near Detector files for *NuMI* stream with S15-02-05 release.

Reconstruction will start at 02:00 p.m.

Crontab list output (novagpvm01.fnal.gov)

```
00 14 * * * source /nova/app/home/novapro/KeepUp_oog/Reconstruction_KeepUp/cron_KeepUp.sh fardet 0
S15-02-05
```

```
00 16 * * * source /nova/app/home/novapro/KeepUp_oog/Reconstruction_KeepUp/cron_KeepUp.sh neardet 0
S15-02-05
```

Notes

* Since November 2014, Near Detector Raw2Root KeepUp Processing is done under FA14-10-03. Due to an error in geometry detector, files from run number = *10377* up to *10620* were reprocessed. A set of raw files have two children files under different releases.

* Since December 12th, 2014, new dimensions for Raw2Root datasets were defined. After an error reported for NOvA production, dimensions were fixed.

2.2.6 Reconstruction Backprocessing

A reconstruction backprocessing request comes usually from the NOvA experiment when a new release with modifications for reconstruction stage is tagged.

Dimensions for datasets that will be used as input of this stage:

```
file_type      = 'importedDetector' and
data_tier      = 'artdaq' and
Online.Detector = '${DET}' and
Online.Partition = 1 and
Online.SubRunStartTime >= '${START_DAY_SECONDS}' and
Online.SubRunStartTime <= '${END_DAY_SECONDS}' and
Online.Stream != 0 and Online.Stream != 2 and not
isparentof:   ( data_tier = 'reco' and
reconstructed.base_release '${RELEASE}')
```

```
*keepup_fd_preshutdown_postshutdown.sh | keepup_nd_preshutdown_postshutdown.sh*
```

This script will submit the jobs to the grid to run reconstruction keepup.

2.3 NOvA Electronic Log Book

NOvA offline software project uses the Electronic Log Book

<http://dbweb3.fnal.gov:8080/ECL/nova/E/index> to record the results from the Keep Up Process.

The screenshot shows a web browser window with the address bar containing the URL: `dbweb3.fnal.gov:8080/ECL/nova/E/create_entry?f=Offline+Keep+Up+Processing`. The page title is "New entry - electronic logbook". The main content area is titled "Create New Entry" and contains the following form elements:

- Form:** A dropdown menu set to "Offline Keep Up Processing" and a "Use" button.
- Category:** A dropdown menu set to "Offline Production" with a "(required)" label.
- Private:** A checkbox labeled "Entry will be visible only to authenticated users".
- Tags:** A text input field containing the following tags: APD, BNEVB, Beam, Block Instrumentation.
- Textile formatted:** A checkbox labeled "Textile help".
- Shifter's Name:** A text input field containing "your_name".
- Online FTS Systems:** A checked checkbox.
- FTS Errors:** A large empty text area.

On the right side of the form, there are instructions and links:

- "Put in your name" next to the Shifter's Name field.
- "Are the following File T Running (must be check domain - Not available)" next to the Online FTS Systems checkbox.
- Links for "Data Disk 1", "Data Disk 2", and "Data Disk 3".
- "Are there errors reported" and "How many? Are they re" next to the FTS Errors field.

A sidebar on the left contains navigation links for Logbook, Shifts, and Admin.

After logging in with their services ID, users click on 'New Entry' on the left, and the pop-up window appears. Form is 'Offline Keep Up Processing', category is 'Offline Production', click 'Private' but you can ignore the 'Tags.' Put your name into the Shifter's Name box and fill in the rest of the information.

2.4 Raw2Root for NOvA

There are six daily cron jobs, divided into two streams. One for far detector, and one for near detector.

- 1 a.m. - far detector NUMI
- 4 a.m. - far detector Cosmics
- 7 a.m. - far detector Others
- 10 a.m. - near detector NUMI
- 12 p.m. - near detector Cosmics
- 2 p.m. - near detector others

Raw2Root has been set up for different streams. Datasets are created for NUMI, Cosmics and Others.

The following picture shows the three steps that make up the raw2root process.

Raw2Root

For i days ago, where $i=1, 3, 5, \dots, 31$



Currently, the cron jobs are called:

1. cron_KeepUp.sh neardet
2. cron_KeepUp.sh fardet

By running these jobs for a date range of the past 1, 3, 5, (every other day for up to 31), there are 16 times that the job is run. In case something goes wrong, there are retries, but the staff also checks on the status of the jobs.

2.4.1 Running the cron job

The cron job is run in novagpvm01, using the group id of novapro. The primary account used for production is the *novapro* account. This account is able to perform submission of special production jobs as well as being the home for a number of production related programs, scripts and items.

For information on production group account, please see:

https://cdcvs.fnal.gov/redmine/projects/novaart/wiki/Production_Group_Account

As of 12/30/2014, the release version for Far detector is S14-08-19. For the Near detector, the release version is FA14-10-03.

A dataset is created automatically each day. Then a sam_web project is created. Then submit the jobs. These three steps are done on data for yesterday, three days ago, and so on going back to thirty-one days ago.

2.4.2 Raw2Root Pattern Dataset Dimensions

file_type = 'importedDetector' and

```

data_tier = 'raw' and not
isparentof: ( data_tier = 'artdaq' ) and
Online.Detector = '${DET}' and
Online.SubRunStartTime >= '${START_DAY_SECONDS}' and
Online.SubRunStartTime <= '${END_DAY_SECONDS}' and
Online.Stream = $STREAM and
Online.TotalEvents > '0'

```

2.5 PChits

The process for running Pchits is similar to raw2root, with creating a dataset, creating a sam_web project, and submitting the job except it done for all files pending up to the current date.

The files produced in this stage are referred to as 'pclist.' These are the files from which the calibration group derives the calibration constants.

There's an script that executes
Cron job and version is:

```

cron_KeepUp_PChits.sh fd S14-08-09
    fd – far detector. Not done for near detector yet.
    It is run every day at 4 a.m.

```

2.5.1 Running the cron job

The cron job is run in novagpvm01, using the group id of novapro. The primary account used for production is the *novapro* account. This account is able to perform submission of special production jobs as well as being the home for a number of production related programs, scripts and items.

For information on production group account, please see:

https://cdcvs.fnal.gov/redmine/projects/novaart/wiki/Production_Group_Account

Until run 17450, will use version S14-08-09, then the version used is S14-09-29.

A dataset is created automatically each day. Then a sam_web project is created. Then submit the jobs. Once the jobs are done, they copy the output files back to an FTS dropbox where a nova FTS instance would copy them to tape and catalogue them in SAM. These steps are done on all data pending to be processed.

2.5.2 Naming format

The names for the PChit datasets follow the format:

“prod_DATATIER_RELEASE_DETECTOR_TRIGGERSTREAM_SPECIAL”

The specific example for fd cosmic stream processed in S14-08-09 would be:
 "prod_pclist_keepup_S14-08-09_fardet_cosmic"

To split this by date, users can add a special field onto the end,
 e.g. for all pchits from events recorded in september 2014:
 "prod_pclist_keepup_S14-08-09_fardet_cosmic_09-2014"

Similarly the pcliststop dataset should be called:
 "prod_pcliststop_keepup_S14-08-09_fardet_cosmic_09-2014"

and the time cal set:
 "prod_timecal_keepup_S14-08-09_fardet_cosmic_09-2014"

As another example, the near detector tricell files (this is called the calibration muon trigger or calmu) would be called:

"prod_pclist_keepup_S14-08-09_neardet_ddcalmu_09-2014"

2.5.3 PCHits pattern

For the current far detector, the PCHits pattern for dataset dimensions

```
data_tier = 'artdaq' and
NOVA.DetectorID = '${DET}' and
Online.Partition = '1' and
Online.Stream = '2' and
Online.Runnumber < '17450' and
Online.SubRunStartTime >= '${START_DAY_SECONDS}' and
Online.SubRunEndTime <= '${END_DAY_SECONDS}' and not
isparentof: (
  data_tier pclist and
  calibration.base_release 'S14-08-09'
)
```

2.6 NOvA Monitoring

These are useful monitoring tools:

FTS:
<http://novasamgpvm01.fnal.gov:8888/fts/status>

<http://novasamgpvm02.fnal.gov:8888/fts/status>

<http://novasamgpvm03.fnal.gov:8888/fts/status>

(Updates continually)

SAM station monitor:

http://samweb.fnal.gov:8480/station_monitor/nova/stations/nova/projects

(Updates when you refresh)

Processing history plots:

<http://nusoft.fnal.gov/nova/datacheck/processinghistory/history.html>

(Updates once a day.)

<http://nusoft.fnal.gov/nova/datacheck/watchdogs/latest/watchdog.html>

Fifemon:

<http://fifemon.fnal.gov/monitor/experiment/nova/batch/>

(Updates when you refresh)

<http://nusoft.fnal.gov/nova/datacheck/watchdogs/latest/watchdog.html>

2.7 Special Situation - Run raw2root manually for NOvA

Occasionally, something happens and there is a request to process files outside of the Keep Up process. The following steps are how this is done when the dataset name is given, or when it has to be created.

1. Log in ssh to novagpvm01.fnal.gov (or the virtual machine you usually use) using the shared group account novapro.
2. If you have your dataset name already, go to step 3.
 - a) If you need to create a dataset, go to NOvA dataset definitions editor
samweb.fnal.gov:8480/sam/nova/definition_editor
 - b) Select Generic Editor
 - c) Select your dataset constraints.

Selection Criteria

Base Data Set:

(To start with a previously defined dataset)

to

(Date format: 2011-05-09 or Date/Time format: 2011-05-09T23:46:04)

- d) Save your dataset, filling in username and group.

Name your dataset: : user: group:

3. Execute:


```
source /nova/app/home/novapro/.bash_profile
export KRB5CCNAME=FILE:/tmp/krb5_novapro
source /grid/ferimapp/nova/novaart/novasun/setup/setup_nova.sh -r {RELEASE} -b maxpot
export X509_USER_PROXY=/var/tmp/novapro.Production.proxy
source submitJob_keepup.sh [DETNAME] [DATASET] [DAYS AGO] [RELEASE]
```
4. Monitor like you do normally to ensure that everything looks all right.

2.8 Condor logs archival for novapro

The production group uses the shared account novapro to submit all production jobs. When using the old jobsub (jobsub_tool), the condor logs (*.out and *.err files) are automatically placed in the folder /nova/ana/condor-tmp/novapro. Sometimes, when an important number of jobs are submitted, the size of this folder can go to the order of TB. It would require someone to wipe out the unused and older condor logs. Due to this, an automatic archival system was set in machine novagpvm02.

The way this automatic system works is by executing two actions.

- 1) Looks for all logs available in nova/ana/condor-tmp/novapro directory which have a modification time older than 7 days and that have a size greater than 1MB. These files are zipped using gzip. Then, all the files that are older than 7 days but newer than 90 days are moved to the folder nova/ana/condor-tmp/novapro_history. They are placed in the folder with matches the day of their modification time.
- 2) Looks for all logs (zipped or not) available in folder nova/ana/condor-tmp/novapro_history which are older than 90 days and remove them.

The archival system consists of a cronjob that executes an script located in the path:
/nova/app/home/novapro/Condor/cron_condor_logs.sh

This script receives one string as the argument. Possible values for the string are 'del' or 'zip'. When the received string is zip, the first step mentioned before is done. When the string is 'del', then the second deletion script is executed.

2.9 novapro Credentials

Normally, users submit jobs using authentication credentials based upon their kerberos login. The novapro account is different. The novapro account uses a special "service certificate" to allow it to more generally submit jobs.

To find up-to-date information on this topic, go to the following redmine page:
https://cdcvns.fnal.gov/redmine/projects/novaart/wiki/Production_Group_Account

3 MINOS

3.1 Script Overview

KEYGEAR is the keepup script and AMBROSIA (KGA) is the batch processing production script used by MINOS, MINOS+ and MINERvA.

The entire job list is submitted as a single entity, meaning that multiple subruns receive a single cluster ID but carry different process numbers. Many functions that used to be done by separate scripts have been wrapped into standard jobsub commands. This allows consolidation of the framework, making it easier to understand the scripts used.

KGA chain



The KGA chain is:

1. keepup_lists --> keepup_list (running samweb versions)
2. KEYGEAR_listbuilder.sh (sets various arguments and passes a final, checked list to the job that runs on the worker node)
3. AMBROSIA_submit.sh (the reconstruction task that runs on the worker node)

3.2 KEYGEAR Script

KEYGEAR_listbuilder.sh is run either by a user to process a list of subruns to be reconstructed manually or as a crontab job for the nightly keepup reconstruction. This script performs bookkeeping and is the start point for submission. Through the careful introduction of flags, this script no longer necessitates submitting each subrun as a single job with the condor_submit commands.

The script is housed in /grid/fermiapp/minos/minfarm/scripts. A detailed list of command line options can be obtained by typing KEYGEAR_listbuilder.sh -h.

Usage: KEYGEAR_listbuilder.sh ID ProcNum VSN iname month pass user do_passes ClusterId

----- HELPERS:

- h - print this list
- E - print list of error codes
- T - test mode - do not update tarfiles or delete daily list
- Z - does not submit and does not write to datalists
- X - write no output
- d - debug options
 - (doesn't remove manually submitted list)

----- POTENTIALLY NECESSARY INPUTS:

- k - use if keepup pass (in crontab)
- l - specifies list name to process (if not keepup, use)
- v - reconstruction version name (e.g. elm2)
- V - second reconstruction version name (useful if testing multiple recos)
- p - pass number (default is -p0)
- j - skip specified number of jobs out of master list (useful for large cron subs)
 - (e.g. if you want to start with line 100, use -j 99)
- r - specify number of files to run out of a list

----- SOFTWARE OPTIONS:

- g - use root compiled with only -g
- O,o - use root compiled with -g -O
- w - run without production role (useful for individual user debug)
- I - integration options
 - (currently exports vars needed for cron; presumed to be a temp measure; if still in use, remember to kinit!)

----- CHECK OVERRIDES:

- b - override bfield check
- B - override beam check
- S - (shutdown) beam off, running cosmic pass only
- U - bypass beam check and run cosmic only -- will suppress ERR 101
- G - bypass beam check but run both passes -- will suppress ERR 101
- L - don't report missing lists -- used when testing
- y - bypass checks, pass -b and -B to submission script = Bbc
- F - force diskpace ignore

----- STREAM OPTIONS:

-A - write all output streams)

-C - write cand output (includes bend)

-N - write ntuple output (includes bntp)

-M - write mrnt output (for spill pass)

----- will default to -N if no input; use -NM for major production run

----- PROCESSING OPTIONS:

-a - do ATMOS processing

-c - do COSMIC processing

-s - do SPILL processing

----- will default to -cs if no input given

----- DETECTOR SELECTIONS:

-n - near det only

-f - far det only

Remember that KEYGEAR_listbuilder.sh can only be run 'for real' on a worker node.

A few of the command line options that are most useful are explained below with examples.

Command line options to KEYGEAR:

1. -v – sets a reconstruction version

For example, -v elm4 runs the elm4 reconstruction. This is required.

2. -l – list of subruns to process

For example, -l testlist. This is required.

3. -k – run in keepup mode.

Optional flag, generally used only during the nightly keepup in the crontab.

It should only be used outside of crontab only by an expert who has full knowledge of how the -k option effects the bookkeeping in the lists/daq_lists directory. If the -k flag is given, the script will look for input lists in the lists/daq_lists directory. It will also append information to various daq lists and tar archives. It does not work with a given, manual list. In fact, if you attempt to use -k and -l, the script will exit without doing anything.

4. -I – integration options that facilitate use with cron with early jobsub scripts.

5. -O – optimized code

Here is an example of setting up the crontab on the appropriate keepup submitting machine to run nightly keepup for elm2 and elm4 reconstructions with optimized code.

```
30 23 * * * . /grid/fermiapp/minos/minfarm/scripts/KEYGEAR_listbuilder.sh -v elm4 -V elm2 -k -O -I
```

This task is running off minos51 user:minospro with a -I flag. The -I sets 'integration options' that facilitate use with cron with early jobsub scripts.

Resubmissions to deal with errors/job crashes are done manually. Also, when a production run is submitted for analysis, this is done by manually submitting a joblist. Lists are specified with the use of the -l flag.

For example:

```
. KEYGEAR_listbuilder.sh -v elm4 -l testlist -O
```

The script will look for testlist in \$LIST_DIR, which is currently assigned to /minos/data/minfarm/lists. The joblist must be located in this directory.

Once KEYGEAR has a list to work with, it will scan to see if the jobs are either currently running, have completed successfully, or match a failed job. The script checks for currently running jobs by looking in \$LIST_DIR/running-jobs for a file that matches the submission. When a job is successfully submitted, a file is created in this directory of the format:

```
D000R#####_S####.PASS.RECO.
```

(e.g. F00061645_0020.0.elm2 for far detector, run 61645, subrun 20, pass 0, with elm2 reconstruction).

If KEYGEAR determines that a subrun in the list is currently running, it will disregard it. Likewise, each subrun is checked against the good_runs and bad_runs lists in \$LIST_DIR for the reconstruction of choice. If the subrun has already been successfully reconstructed, it will also be disregarded. If the subrun is found in the bad_runs list, that job will be submitted to the grid, but it will give a warning to remove that subrun from the bad_runs list.

After performing these checks, KEYGEAR stores the subruns that passed the cuts in a file located in \$LIST_DIR/KGA/. These files are identified as (manual/keepup).timestamp. Note that these files must be kept in the directory until the jobs are completed, as AMBROSIA will copy specified lists to the grid nodes to actually perform reconstruction.

KEYGEAR additionally sets up the arguments for the AMBROSIA_submit script depending upon the flags given. It then communicates with the grid nodes through the jobsub command, which means that we have reconstruction jobs running on worker nodes.

3.3 AMBROSIA Script

AMBROSIA_submit.sh is the actual reconstruction job running on the worker node. It creates ntuples of choice as well as DSTs for data quality purposes. It takes the arguments provided by KEYGEAR_listbuilder.sh and adjusts accordingly. It also removes the associated file in \$LIST_DIR/running-jobs, indicating that the job has finished one way or another, and updates the good_runs & bad_runs lists of the relevant reconstruction version.

3.4 Useful Information

By default KGA will run both cosmic and spill passes and produce standard ntuples only. This works well during nightly keepup. However, during an analysis production run, we generally want both standard ntuples and muon-removed ntuples. For the elm5 production run, the submission would be:

```
. KEYGEAR_listbuilder.sh -v elm5 -l giantelm5list -ONM
```

where N and M specify both sntp and mrnt files should be produced.

Remember to source `/grid/fermiapp/minos/scripts/jobsub.sh` to use the new jobsub tool. If the cluster ID number looks odd after submitting a job, it is likely that the jobs have gone into the wrong pool and will crash. This might necessitate manual cleanup of the running-jobs directory, so take care with this step.

It's also worth noting that care must be taken in submitting FD jobs to avoid overloading the database. These jobs generally run quickly. In the past, a cron job was used to partition the FD list; however, that functionality does not currently exist in the KGA framework. The `-j` and `-r` flags exist, which allow the user to skip a certain number of lines in a large list and set the number of subruns to submit respectively. A new wrapper will be written to make use of these flags and handle this issue in the future.

3.5 Minos Monitoring

Various current plots are linked from the MINOS data handling page at:

<http://www-numi.fnal.gov/computing/dh/dhmain.html>

The Minos at Work page requires special access.

<http://www-numi.fnal.gov/Minos/>

The Minos Control Room page requires special access.

<http://www-numi.fnal.gov/Minos/ControlRoom/index.html>

You may request access at: <http://www-numi.fnal.gov/sitefeedback.html>

4 MINERvA

4.1 Monte Carlo Production

Flux files are fed to GENIE. Currently, these files reside on BlueArc. As MINERvA begins to use CVMFS, we will pursue the associated solution for large auxiliary file distribution. The output of the GENIE simulation is fed to MINERvA's Geant4-based detector simulation. This simulation includes “calibrations”. Afterwards we generate MINOS detector simulation using data overlay from MINOS

and the exiting particle spectra from MINERvA. Subsequently we run reconstruction.

4.2 Data Production

After keep-up processing, the entire chain of production for data uses dCache.

4.3 User Analysis

User analysis jobs ultimately use the bulk of our grid computation time. In many cases, user analysis will be beyond the scope of duty for the Production Ops team. Some interesting possible exceptions are when the collaboration decides to produce “official releases” of some user analysis products (either for use by multiple analyzers or to freeze files for publications) and test jobs that ensure the user analysis portion of the production chain are working well and in good compliance with best practices.

5 Production Tools

The following tools are used in Production.

- SAM
- FTS
- Jobsub
- Monitoring

6 Glossary

Calibration – the phase where computations are done to determine correction factors for measurement errors in the detector(s). This is usually a statistical analysis of each data channel in the detector over a batch of data to compare that channel's values to an expected distribution, and figure a correction factor based upon that. The calibration values are often stored in a conditions database to be used to correct data channel by channel for reconstruction. The calibration can also declare detector channels dead, or noisy (disconnected).

HTcondor - Batch scheduler system (<http://research.cs.wisc.edu/htcondor/>) used in a two-layered mode with a package called Glideinwms, where we schedule "glideins" (http://research.cs.wisc.edu/htcondor/manual/v7.6/5_4Glidein.html) (which are jobs acting as batch-node-clients) to GRID nodes. The glide-ins advertise batch slots to our batch system, and then we submit jobs to those glide-in batch slots, a batch system within a batch system. This way, the "glide-ins" can do things like renew GRID credentials, enforce various limits and quotas, and provide other useful services that a plain GRID slot would not.

FTS- The File Transfer System (FTS) handles cataloging, replicating, archiving and deleting files. The FTS interfaces with the SAM metadata catalog and data delivery systems and supports interactions with the Enstore tape archive facilities as well as Bluearc central storage. It watches one or more Dropbox directories, and puts files which appear there away, according to a configuration file. This usually includes declaring files to SAM, and verifying they were written to tape before removing them from the Dropbox.

FTS Dropbox - The File Transfer System (FTS) storage in the cloud. In UNIX terms, a dropbox is just a directory with an FTS instance monitoring that dropbox. The role of the FTS is to declare (check-in) files to SAM and transfer them to their final locations. In other words, the dropboxes allow production to drop off the files and have FTS handle the rest.

jobsub tools - used to submit jobs to HTCondor pools. Various input options steer jobs to the different pools and control input and output of data files to the user application running on the worker nodes.

KeepUp – any processing that needs to be done on raw data as it comes in -- to "keep up" with the new data. This can include calibration, format conversion (raw2root), and other miscellaneous accounting. (<http://minerva-docdb.fnal.gov/cgi-bin/RetrieveFile?docid=9793>)

Pchits – the processing of files to feed into the Reconstruction phase.

Production Operations– the overall process of going from raw files to information that can be used by physicists for analysis. This includes keep-up, calibration, format conversion (raw2root), and reconstruction, as well as monte-carlo simulation data generation.

Raw2Root –format conversion, the processing of raw files into files in the ROOT Data Analysis framework. This may be called Raw2digits in another system.

SAM - Sequential Access via Metadata (SAM) is a data handling system to store and retrieve files and associated metadata, including a complete record of the processing which has used the files.

SAM Dataset - a metadata query that resolves to a desired set of input files for processing. Dataset

definitions may be created by SAM client tools on the command line or with experiment-specific web-based tools. Users build a list of metadata parameters to define a collection of like files, the dataset.

SAM Metadata - data about the data, used to specify the properties of the files. For example: `data_tier = reconstructed` (only want reconstructed files), or `nova.detectorid = fd` (only want far detector files).

SAM Project - SAM component that deals out files from a snapshot of a SAM dataset to jobs as they become available, and also tries to pre-stage files that need to be staged in from tape. A project represents the pool of files available for delivery and the collection of Processes pulling files from that pool. The Project is typically started by specifying the dataset definition name, and the SAM snapshot is created at that time. The name of the SAM Project is also specified. Internal to SAM, a Project is a unix process called *pmaster* running on the SAM station node. *pmaster* responds to requests from SAM processes and coordinates the delivery of files particular to the Project.

7 Appendix A

Renaming wrong named files.

Due to a DAQ bug, a small set of raw files were wrongly named. Those files ended with the name pattern: "`_[detector_name]%.raw%.raw_`"

Following instructions from Andrew Norman, those files were retired from SAM and new files with the right name were declared. Those files were also renamed on tape (novasamgpvm01.fnal.gov).

The list of renamed files are available at:

- <https://cdcv.s.fnal.gov/redmine/attachments/download/23037/FDRenamedFiles.txt>
- <https://cdcv.s.fnal.gov/redmine/attachments/download/23038/NDRenamedFiles.txt>