

Weekly Meeting

April 18th 2018

Visa Application

- I have my appointment at US consulate tomorrow, and I have all my documents.
- If approved, I will be able to book my flights
- I will also ask about going through this process again

Supplement Xsec Doc

- Paper was not submitted last week
- I'm just responding to comments as they come in about the supplement doc

Now focusing on other work

- Timing for noisePSD for Amy
- Working on DQM
 - Build more deciders
 - Work on alarm system
- IO Library
 - Ben has put the midas reading/writing in IO Lib
 - I can finish working this in to DMC
- Dark photon projected limits for SNOLAB and CUTE

Noise PSD Timing Analysis

- Having trouble getting cdmsbats to compile

3. Compiling CDMSBats - An optional makefile method

This section describes an *optional* way to compile the packages that reside within cdmsbats. From this directory, users may build any or all of the job stages for raw data processing: BatCalib, BatRoot, BatNoise, or Batviewer, along with the BatCommon library of shared facilities for all job stages.

Note that ROOT v5.xx.xx (compiled with FFTW), Python 2.x or newer, Boost, and BLAS libraries must all be installed to compile the above executables.

For users who work with all the stages of processing, or who are unsure what they might need to run, everything can be built in one step. From the cdmsbats top level directory, type:

```
$ make all
```

If you just need one of the executables, you can simply use the directory for the component you wish to install as the "make" target, for example:

```
$ make BatRoot
```

The make command places the executables into the BUILD/bin directory. To run a package without having to specify the full path to this directory, you may set your path to point to this directory, depending on your shell:

```
TCSH: $ setenv PATH ${PATH}:/path/to/cdmsbats/BUILD/bin
```

```
BASH: $ export PATH=${PATH}:/path/to/cdmsbats/BUILD/bin
```

Now you should be able to run the target executable from anywhere by simply typing, as an example:

```
$ BatNoise 01111111_0001 1 1000
```

Noise PSD Timing Analysis

- Compiling seems simple, if you have the correct environment variables.
- I'm trying to do it on nero, since it gives the environment variables needed on nero as an example:

```
#!/bin/tcsh
#
#This is an example script defining the paths and environment variables
#required to run BatNoise to process a set of raw data.
#
#Required library paths:
setenv ROOTSYS /nerosoft/root/
setenv PYTHONPATH /galbadata/analysis/packages/anaconda2/lib
setenv PATH ${PATH}:${ROOTSYS}/bin:/galbadata/analysis/packages/anaconda2/bin:/scratch/danika/cdmsbats/BUILD/bin:.
setenv LD_LIBRARY_PATH ${ROOTSYS}:${PYTHONPATH}:/galbadata/analysis/packages/boost_1_65/lib
#
#BatNoise required environment variables.
setenv CDMSBATSDIR /scratch/danika/cdmsbats
setenv BATROOT_CONST /scratch/josh/cdmsbats/UserSettings/BatRootSettings/analysis
setenv BATROOT_PROC /scratch/josh/cdmsbats/UserSettings/BatRootSettings/processing
setenv BATNOISE_TEMPLATES /scratch/josh/cdmsbats/PulseTemplates/
setenv BATROOT_NOISEFILES /scratch/josh/cdmsbats_test_folder/NOISEFILES
setenv BATROOT_RQDATA /scratch/josh/cdmsbats_test_folder/RQDATA
#
#The following are run specific and should be adjusted as needed:
setenv BATROOT_RAWDATA /data/R133/raw/01140214_2214
setenv BATROOT_AUXFILES /data/R133/raw/01140214_2214
setenv BATROOT_GPIBFILES /data/R133/raw/01140214_2214
```

