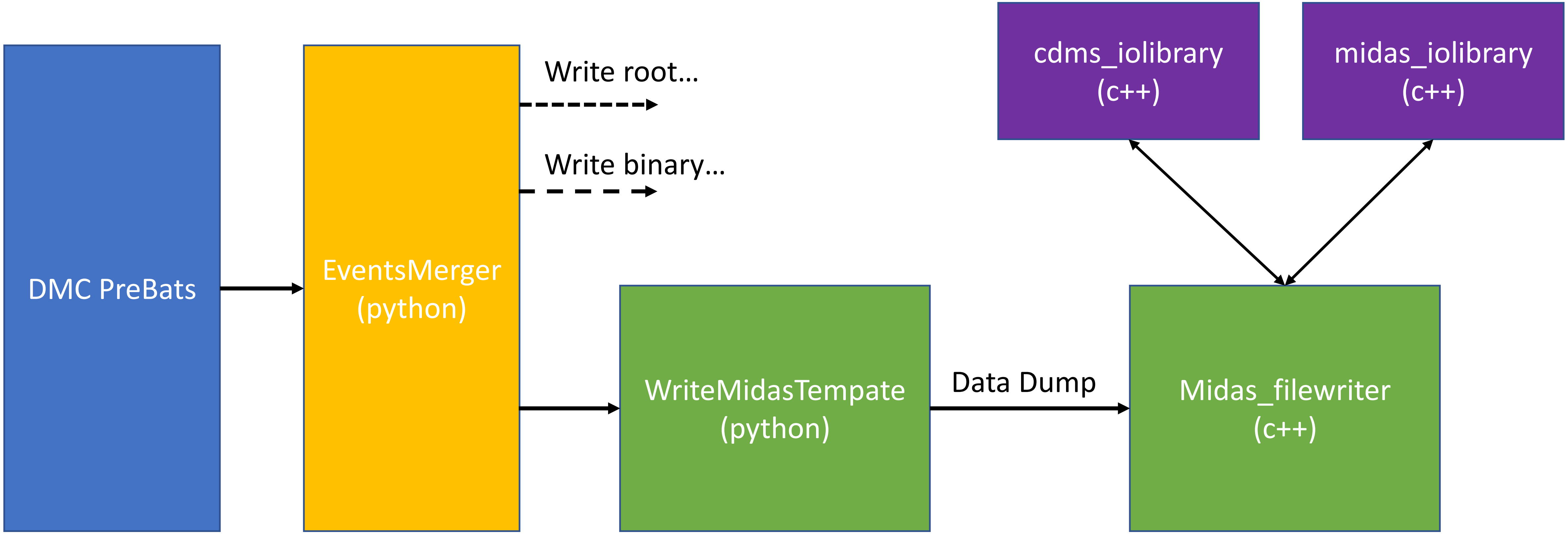
Weekly Meeting

September 7 2017

DMC – MIDAS file writing code

- General code structure:



DMC – MIDAS file writing code

- Reasons for this structure:
  - Keeping libraries in c++ environment, keeps all functionality and mitigates issues with wrapping
    - I can get cdms_iolibrary to wrap using a generator, but not sure if midasio will work as easilty
  - Building: I can create a very simple wrapper using boost/python, which is available on brazos already, which makes it easier to people to compile code.
    - Before I was using SWIG to generate code, which isn’t installed on brazos cluster
  - Can use the simple wrappers to “dump” the DMC data from python code to c++ code

DMC – MIDAS file writing code

- Downsides to this structure:
  - It’s likely not the most efficient way to do things, but it will work.
  - Why not link directly to libraries (instead of data dump)?
    - Again, building without something like SWIG
    - Worried about wrapping midasio
    - Wrapping with boost/python becomes very complicated (python data are in complex structures, like python lists of arrays, 4D objects)
  - Could implement a more efficient method, but I want to get out something that works

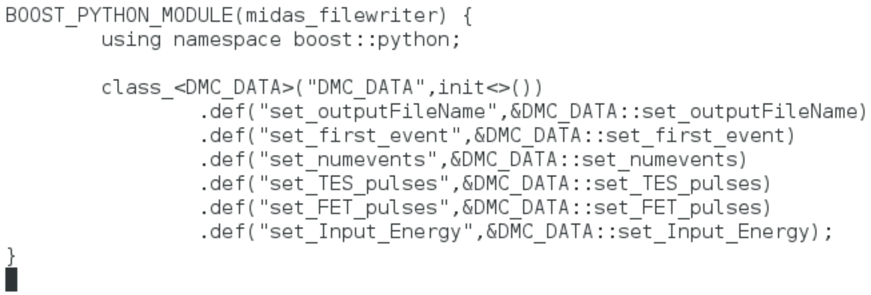
DMC – MIDAS file writing code

- The data dump – I’m transferring the DMC data into c++ objects that I can use with the libraries
- E.g., pulse data will be put into 4D vector in python (manually)
- SIM data will be put into a data struct

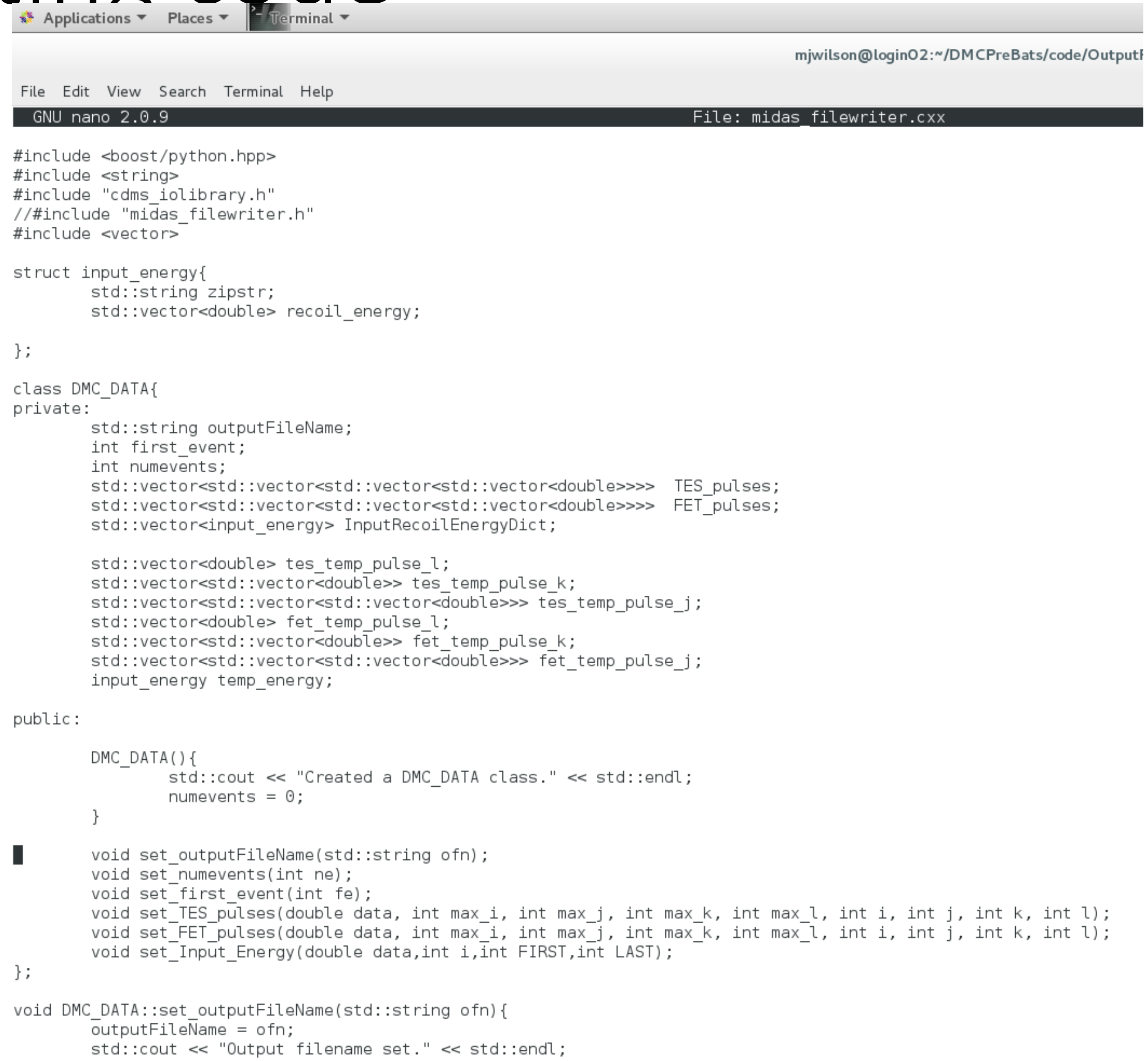
- DMC data will be loaded into a c++ class in midas_filewriter.cxx (using set functions)
- Function within this c++ class will take data and write to (one) MIDAS file

DMC – MIDAS file writing code

- It’s messy, but it works

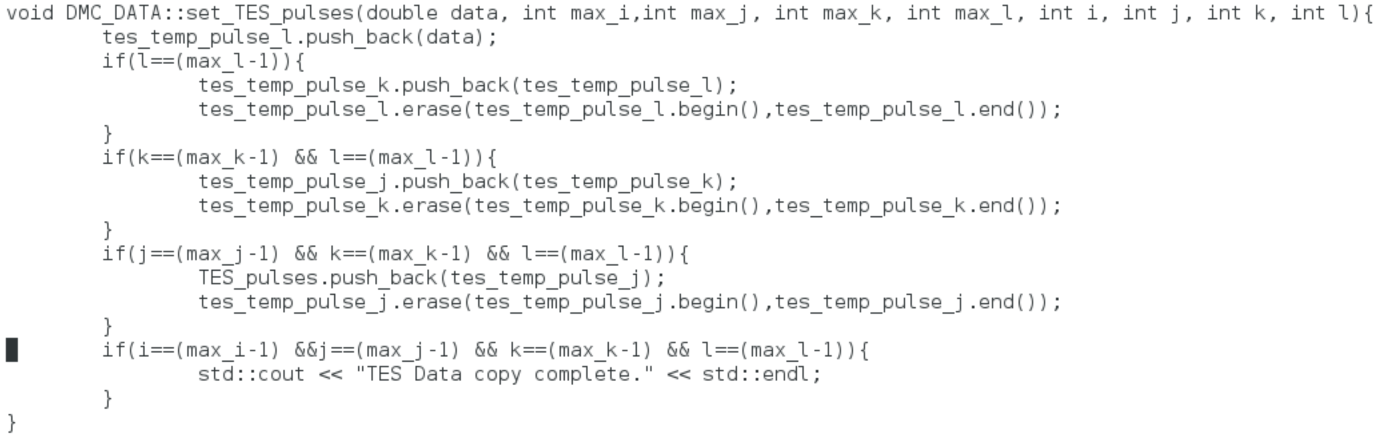


```
BOOST_PYTHON_MODULE(midas_filewriter) {
        using namespace boost::python;

        class_<DMC_DATA>("DMC_DATA",init<>())
                .def("set_outputFileName",&DMC_DATA::set_outputFileName)
                .def("set_first_event",&DMC_DATA::set_first_event)
                .def("set_numevents",&DMC_DATA::set_numevents)
                .def("set_TES_pulses",&DMC_DATA::set_TES_pulses)
                .def("set_FET_pulses",&DMC_DATA::set_FET_pulses)
                .def("set_Input_Energy",&DMC_DATA::set_Input_Energy);
}
```

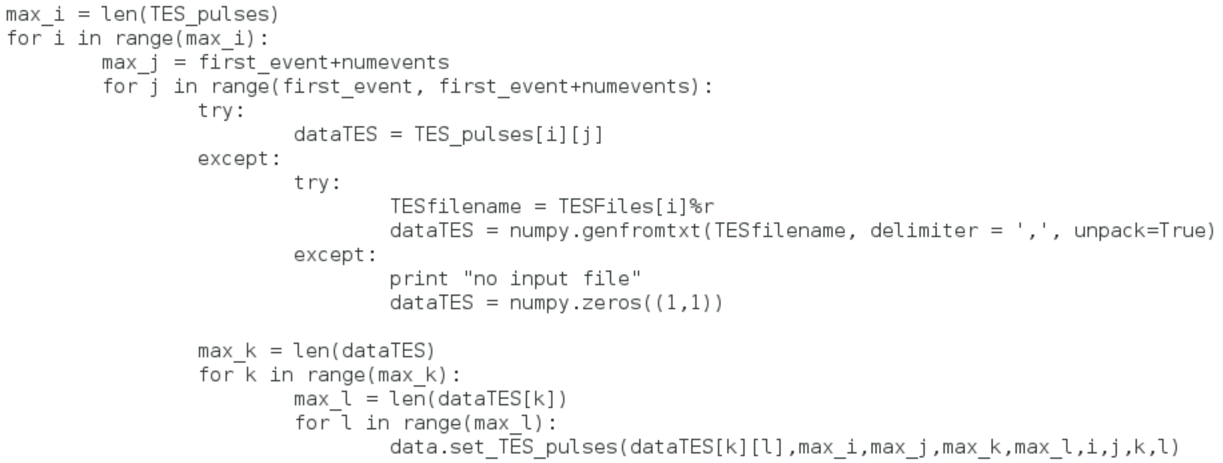

Applications
Places
Terminal
mjwilson@login02:~/DMCPreBats/code/OutputF
File Edit View Search Terminal Help
GNU nano 2.0.9
File: midas_filewriter.cxx

```
#include <boost/python.hpp>
#include <string>
#include "cdms_iolibrary.h"
//#include "midas_filewriter.h"
#include <vector>

struct input_energy{
        std::string zipstr;
        std::vector<double> recoil_energy;

};

class DMC_DATA{
private:
        std::string outputFileName;
        int first_event;
        int numevents;
        std::vector<std::vector<std::vector<std::vector<double>>>>  TES_pulses;
        std::vector<std::vector<std::vector<std::vector<double>>>>  FET_pulses;
        std::vector<input_energy> InputRecoilEnergyDict;

        std::vector<double> tes_temp_pulse_l;
        std::vector<std::vector<double>> tes_temp_pulse_k;
        std::vector<std::vector<std::vector<double>>> tes_temp_pulse_j;
        std::vector<double> fet_temp_pulse_l;
        std::vector<std::vector<double>> fet_temp_pulse_k;
        std::vector<std::vector<std::vector<double>>> fet_temp_pulse_j;
        input_energy temp_energy;

public:

        DMC_DATA(){
                std::cout << "Created a DMC_DATA class." << std::endl;
                numevents = 0;
        }

        void set_outputFileName(std::string ofn);
        void set_numevents(int ne);
        void set_first_event(int fe);
        void set_TES_pulses(double data, int max_i, int max_j, int max_k, int max_l, int i, int j, int k, int l);
        void set_FET_pulses(double data, int max_i, int max_j, int max_k, int max_l, int i, int j, int k, int l);
        void set_Input_Energy(double data,int i,int FIRST,int LAST);
};

void DMC_DATA::set_outputFileName(std::string ofn){
        outputFileName = ofn;
        std::cout << "Output filename set." << std::endl;
```

DMC – MIDAS file writing code



```
void DMC_DATA::set_TES_pulses(double data, int max_i,int max_j, int max_k, int max_l, int i, int j, int k, int l){
        tes_temp_pulse_l.push_back(data);
        if(l==(max_l-1)){
                tes_temp_pulse_k.push_back(tes_temp_pulse_l);
                tes_temp_pulse_l.erase(tes_temp_pulse_l.begin(),tes_temp_pulse_l.end());
        }
        if(k==(max_k-1) && l==(max_l-1)){
                tes_temp_pulse_j.push_back(tes_temp_pulse_k);
                tes_temp_pulse_k.erase(tes_temp_pulse_k.begin(),tes_temp_pulse_k.end());
        }
        if(j==(max_j-1) && k==(max_k-1) && l==(max_l-1)){
                TES_pulses.push_back(tes_temp_pulse_j);
                tes_temp_pulse_j.erase(tes_temp_pulse_j.begin(),tes_temp_pulse_j.end());
        }
        if(i==(max_i-1) &&j==(max_j-1) && k==(max_k-1) && l==(max_l-1)){
                std::cout << "TES Data copy complete." << std::endl;
        }
}
```



```
max_i = len(TES_pulses)
for i in range(max_i):
        max_j = first_event+numevents
        for j in range(first_event, first_event+numevents):
                try:
                        dataTES = TES_pulses[i][j]
                except:
                        try:
                                TESfilename = TESFiles[i]%r
                                dataTES = numpy.genfromtxt(TESfilename, delimiter = ',', unpack=True)
                        except:
                                print "no input file"
                                dataTES = numpy.zeros((1,1))

                max_k = len(dataTES)
                for k in range(max_k):
                        max_l = len(dataTES[k])
                        for l in range(max_l):
                                data.set_TES_pulses(dataTES[k][l],max_i,max_j,max_k,max_l,i,j,k,l)
```

DMC – MIDAS file writing code

- I have now connected with Jorge again
- We have set up a google doc for me to write down all of my questions I have gathered, and he will answer them
- No updates on midasio status – Thomas said he will talk to midas developers when he gets back to Vancouver

QM Work

- Document for Soudan deciders has not been updated with the Type 3 or Type 4 deciders yet
  - These are the deciders that will involve some level data processing
  - These deciders will most likely be the ones that I will work on
- Should I contact Amy to see if those deciders can be added?
- Is it worth looking at other deciders at all?

Dark Phonon

- Will look into photoelectric absorption cross sections in G4
    - Best way to get started?