Weekly Meeting

July 26 2017

It’s been a quiet week

- Have not heard back from either Belina or Amy about the status of IO Library in regards to MIDAS dependency
  - Whether or not a decision has been made
  - If not, what will the decision depend on?
- I asked specifically about whether there has been a decision made, and timeline of IO Library implementation into DAQ – got response from Ben:
  - I think it should only take a day or so to add IOLibrary to the rest of the DAQ. The library is working very well in the event builder, so I don't expect too many teething troubles (you may have seen a few pushes from me to fix some edge cases already).
  - From a technical perspective, implementing IOLibrary in the DAQ shouldn't have any effect on the decision about whether to add a MIDAS dependency to IOLibrary.

It’s been a quiet week

- I am still hoping to have a meeting with Belina, new Ben, Amy, other Ben, about how the implementation has been going, what changes have been made, how to move forward.

- I am also thinking about giving a talk during the next DMC meeting – talk about the status of binary writer/io library
  - The binary writer for the DMC is being held up by the IO Library, as we need to figure out those issues first.
  - Maybe it is a good idea to officially say this, and let them know what decisions are being made, and how it affects the DMC?

C++/python wrapper

Python does not like pointers!!

C++/python wrapper

- I had a lot of trouble getting pointers to work in python. For instance:

```
struct my_struct{
int a;
int *b;
};
```

```
>>> import example
>>> t = example.my_struct()
>>> t.a = 5
>>> t.b = ???
```

C++/python wrapper

- I had a lot of trouble getting pointers to work in python. For instance:

```
struct my_struct{
int a;
int *b;
};
```

```
>>> import example
>>> t = example.my_struct()
>>> t.a = 5
>>> t.b = ???
```

C++/python wrapper

- It took me a while to figure out how to get pointers to work in python → and thus getting arrays to work as I need them in python
  - Internet searching was mainly concerned with passing pointers into functions, which I didn't have a problem with
- Eventually I found some macros(?) in SWIG that help with this

C++/python wrapper

```
#include <iostream>
#include "example.h"

void add(ptr_struct *my_ptr, int *array_ptr){
        my_ptr->c = my_ptr->a + my_ptr->b;
        my_ptr->result = &(my_ptr->c);
        array_ptr[0] = my_ptr->a;
        array_ptr[1] = my_ptr->b;
        array_ptr[2] = my_ptr->c;
        array_ptr[3] = (my_ptr->data[0]);
        array_ptr[4] = (my_ptr->data[1]);
        array_ptr[5] = (my_ptr->data[2]);
}
```

```
#ifndef EXAMPLE_H
#define EXAMPLE_H

#include <iostream>

struct ptr_struct {
        int a;
        int b;
        int c;
        int *data;
        int *result;
};

void add(ptr_struct *my_ptr, int *array_ptr);


#endif
```

C++/python wrapper

```
%module example
%include "cpointer.i"
%include "carrays.i"

%pointer_functions(int,intp);
%array_functions(int,intArray);

%{
#include "example.h"
%}

%include "example.h"
```

```
[mwilson@cdms pointer_test]$ python
Python 2.7.5 (default, Nov  6 2016, 00:28:07)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-11)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import example.ptr_stuct()
  File "<stdin>", line 1
    import example.ptr_stuct()
                           ^
SyntaxError: invalid syntax
>>> exit()
[mwilson@cdms pointer_test]$ python
Python 2.7.5 (default, Nov  6 2016, 00:28:07)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-11)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import example
>>> t = example.ptr_struct()
>>> t.a = 2
>>> t.b = 17
>>> data = example.new_intArray(3)
>>> example.intArray_setitem(data,0,9)
>>> example.intArray_setitem(data,1,24)
>>> example.intArray_setitem(data,2,5)
>>> t.data = data
>>> array = example.new_intArray(6)
>>> example.add(t,array)
>>> example.intArray_getitem(array,0)
2
>>> example.intArray_getitem(array,1)
17
>>> example.intArray_getitem(array,2)
19
>>> example.intArray_getitem(array,3)
9
>>> example.intArray_getitem(array,4)
24
>>> example.intArray_getitem(array,5)
5
>>>
```

- Can create an array in python
- Can pass array in function to be “filled”
- Can have pointer declared in c++ code be set to array created in python

C++/python wrapper

- So now, I can get python code to work with vectors, pointers, and arrays from c++ code!
- This is covers much of the functionality in the IO Library
- Need to handle the little things to get entire IO Library to work with python

UBC/TRIUMF to Chicago

- I leave next week
  - Will be gone Thursday and Friday
- I will be at UBC/TRIUMF the following week, after long weekend
  - Be with Thomas, new Ben and mysterious new Andrew
  - Mostly be at TRIUMF
- Week after, I head to Chicago for collaboration meeting