

## MAUS - Support #602

### UploadGeometry.py - Code Review

29 July 2011 16:45 - Littlefield, Matthew

|  |                     |                        |  |
|--|---------------------|------------------------|--|
| <b>Status:</b>   | Rejected            | <b>Start date:</b>     | 29 July 2011   |
| <b>Priority:</b>   | Normal              | <b>Due date:</b>       |  |
| <b>Assignee:</b>   | Rogers, Chris       | <b>% Done:</b>         | 0%   |
| <b>Category:</b>   | common_py           | <b>Estimated time:</b> | 0.00 hour  |
| <b>Target version:</b>   | Future MAUS release |                        |  |
| <b>Workflow:</b>   |                     |                        |  |
| <b>Description</b>   |                     |                        |  |
| In my development branch is UploadGeometry.py which is the executable to upload to the CDB and it uses a few different classes. Could you review it please? Tests for the classes will be along soon :-) |                     |                        |  |
| <b>Related issues:</b>   |                     |                        |  |
| Related to MAUS - Support #697: Code Review - Geometry Handling system   |                     | <b>Closed</b>          | <b>12 September 2011</b> <del>2011</del> <b>September 2011</b> |

#### History

##### #1 - 01 August 2011 15:10 - Rogers, Chris

- Category set to common\_py
- Target version set to Future MAUS release

You mean uploadGeometry.py? Any tests to go with it?

##### #2 - 02 August 2011 10:20 - Littlefield, Matthew

I've changed it to UploadGeometry.py its in /maus-littlefield/src/common\_py/geometry

and the command line argument it requires is Config.txt

I was going to work on the tests once you were happy with the flow of the executable through the classes. I can sort out the tests today if you like?

The tests for GDMLFormatter.py and GDMLtoCDB.py need a little more thinking done because they are unusual I was going to look at them in more depth once the Download executables had been written this week.

##### #3 - 02 August 2011 10:46 - Littlefield, Matthew

I found a runtime error yesterday in the code I uploaded on Friday which didn't flag errors which was fixed yesterday and just now so the working version of UploadGeometry.py is in my branch revision 596 :-)

##### #4 - 02 August 2011 11:40 - Rogers, Chris

I mean, I don't really know what you want me to review. UploadGeometry.py fails pylint, has no docstring, and no tests right? Is that really the code you want reviewed? Maybe I'm looking at the wrong version?

##### #5 - 02 August 2011 13:41 - Littlefield, Matthew

To begin with I wanted to make sure you were happy with,

- The way the executable has Config.txt as its command line argument and the contents of Config.txt are ok?
- The flow of the executable through the classes and the way the classes interact? Could this be done better?
- The classes it calls and the way in which these classes are written. Do they need to be broken down into smaller methods? Are they too confusing?
- Have I made any glaring mistakes?
- Are there any specific tests you would like to see?
- Any other general comments?

In the meantime I'll make sure they pass pylint and are commented correctly and that the tests work and that there are tests for the CDB calling class.

##### #6 - 02 August 2011 15:59 - Rogers, Chris

Okay, my misunderstanding. When we talked last week, you indicated that the code would be ready to role by Friday. So I was expecting polished code.

Config.txt - I thought we agreed you would use the common Configuration pattern, as in \$MAUS\_ROOT\_DIR/src/common\_py/Configuration.py  
Executable flow - I think this is okay.

Classes it calls - okay. Obviously need documentation.

Other stuff:

I can answer once you have finished writing the code, tests, documentation. If the tests aren't thorough enough I'll ping you!

**#7 - 03 August 2011 14:23 - Littlefield, Matthew**

My apologies I should have been clearer I meant the raw code would be finished i.e. you would be able to upload geometries to the CDB. I've now gone through the code and classes with pylint and there are few messages on there I would like to disable. I know you can do this with #pylin: disable = "number" but I'm unsure where to get the number from? Also they should all be properly documented now as well.

I have also written a unit test for GDMLPacker.py which is in the devel branch and works. The other 3 classes in the executable are a little trickier to write and I need to take a further look at them and perhaps talk with you about the next time I come to RAL. Which leads me to my next question. Can I come up on Monday or Tues? :)

Also have some questions on the Configuration.py file which we can talk about the next time I come up if that's ok?

Finally, as per the deadlines I said last week I intend to complete the executables which download from the database by Friday but not quite in polished form yet as I would like to complete the workflow loop then do the polishing. I hope this is all ok?

**#8 - 03 August 2011 14:35 - Tunnell, Christopher**

pylint -i y FILENAME.py

**#9 - 09 August 2011 01:01 - Tunnell, Christopher**

I volunteer myself as the test dummy to go through and make sure everything is clear since I have no idea how this all works and would like to. Let me know when it gets to that point :)

**#10 - 09 August 2011 09:05 - Littlefield, Matthew**

OK sounds good to me I'll keep you posted

**#11 - 19 October 2011 11:27 - Rogers, Chris**

- Status changed from Open to In Progress

- % Done changed from 0 to 50

I'll start going through this stuff today...

**#12 - 15 November 2011 05:22 - Rogers, Chris**

- Status changed from In Progress to Rejected

- % Done changed from 50 to 0

Reject - will be handled in [#697](#)