

BornAgain - Feature #1563

Feature # 1290 (Rejected): === Core: framework ===

Feature # 1572 (Rejected): == Python ==

Investigate time-of-life of SampleBuilder in Python context

03 Aug 2016 08:48 - pospelov

| | | | |
|--|----------|------------------------|-------------|
| Status: | Archived | Start date: | 03 Aug 2016 |
| Priority: | Normal | Due date: | |
| Assignee: | herck | % Done: | 0% |
| Category: | | Estimated time: | 0.00 hour |
| Target version: | Sprint32 | | |
| Description | | | |
| Item 1 | | | |
| <ul style="list-style-type: none">• Is it possible to provide inline usage of SampleBuilder, as in example of Dominique Dresen? | | | |
| Crash if SampleBuilder object is not referenced in python script | | | |
| If I write a SampleBuilder class, say SphereMonolayer(ba.ISampleBuilder), is there some specific reason why I have to initiate it as an object and can not directly pass it to setSampleBuilder of the GISASSimulation object? What I mean: | | | |
| <pre>samplebuilder = SphereMonolayer() simulation.setSampleBuilder(samplebuilder)</pre> | | | |
| is fine but | | | |
| <pre>simulation.setSampleBuilder(SphereMonolayer())</pre> | | | |
| leads to Aborted(core dumped). Otherwise the variable samplebuilder would not be used within my code. I guess this is some memory problem? | | | |
| The problem here is that Python garbage collector removes SphereMonolayer right after setSampleBuilder call, thus not giving C++ a chance to access it later. However, SampleBuilder should be passed inside C++ in the form of shared pointer. Why it doesn't work? | | | |
| Item 2 | | | |
| In Python sample builder examples we show, that buildSample method returns a newly created sample, and to prevent the deletion of this sample by Python we store it in the class variable | | | |
| <pre>def __init__(self): self.sample = None</pre> | | | |
| <pre>def buildSample(self): self.sample = ba.MultiLayer() return self.sample</pre> | | | |
| Additionally, we have weird code in Simulation::updateSample | | | |
| <pre>if(builder_type.find("ISampleBuilder_wrapper")</pre> | | | |
| Which looks like attempt to guess, if SampleBuilder came from Python and adjust ownership according to it. This piece of code looks like remnants of boost-python bindings, I think it is never executed. | | | |
| Within this item: | | | |
| <ul style="list-style-type: none">• Make sure we understand what is going on with SampleBuilder and it's sample on the way from Python to C++ | | | |

History

#1 - 03 Aug 2016 09:51 - jmfisher

This is definitely a remnant from the old bindings. In the old bindings, the ownership problem was solved on the C++ side by checking whether the sample builder comes from Python or C++, and if from Python it creates a clone of the sample builder to bypass ownership problems. With the new bindings, we could implement exactly the same behaviour with swig %extend directives. The two methods which need to be extended are Simulation::updateSample and SpecularSimulation::updateSample

#2 - 03 Aug 2016 19:08 - wuttke

- Parent task set to #1572

#3 - 20 Oct 2016 13:12 - herck

- Assignee set to herck

#4 - 26 Oct 2016 17:48 - herck

Item 2 is also solved

#5 - 27 Oct 2016 15:42 - herck

- Status changed from Sprint to Resolved

All different use cases should now be possible without issues regarding object ownership. With many thanks to our swig expert Jonathan for all the help!

#6 - 11 Nov 2016 14:48 - herck

- Status changed from Resolved to Archived