

MAUS - Bug #1832

Interesting feature in TOF data structure when accessed from python

16 March 2016 09:32 - Rogers, Chris

Status:	Open	Start date:	16 March 2016
Priority:	Normal	Due date:	
Assignee:	Rajaram, Durga	% Done:	0%
Category:	TOF	Estimated time:	0.00 hour
Target version:	Future MAUS release		
Workflow:	New Issue		

Description

Here is an interesting feature in the TOF data structure. The attached script generates toy TOF0, TOF1, TOF2 space points and puts them on a TOF Event. The script then pulls them off the TOF event and prints them. Features which look fishy to me:

1. the TOF1 and TOF2 events when read have the same memory address
2. All of the TOF space points have the same member data

TOF data structure stores everything as a value (no pointers involved as far as I can see). The SetBlah functions should be doing a copy operation, so I don't see how this is possible.

For information I am trying to set up a test for the global track fit, so I want to generate dummy data.

History

#1 - 16 March 2016 20:38 - Rajaram, Durga

- File `test_tof_sp.cc` added
- File `Makefile.testtof` added
- File `cpp_test_tof_sp.log` added

I'm not sure if there's some pythonic pointer handling going on behind the scenes

```
All of the TOF space points have the same member data
```

Are you referring to the "third" get? If so, that should be expected since `tof_sp` is basically what you have from the last "set" from within the

```
for station, z in enumerate(..)
```

loop?

I took your python test and turned into a c++ program and I don't see those features attached cc, make, log

Files

<code>test_tof_sp.log</code>	995 Bytes	16 March 2016	Rogers, Chris
<code>test_tof_sp.py</code>	2.23 KB	16 March 2016	Rogers, Chris
<code>test_tof_sp.cc</code>	4.18 KB	16 March 2016	Rajaram, Durga
<code>Makefile.testtof</code>	330 Bytes	16 March 2016	Rajaram, Durga
<code>cpp_test_tof_sp.log</code>	624 Bytes	16 March 2016	Rajaram, Durga