MAUS - Bug #1716

Exceptions occasionally not getting caught on SL5

20 July 2015 04:52 - Rajaram, Durga

Status:	Open	**Start date:**	20 July 2015
Priority:	Normal	**Due date:**	
Assignee:	Dobbs, Adam	**% Done:**	0%
Category:	Data Structure	**Estimated time:**	0.00 hour
Target version:	Future MAUS release		
Workflow:	New Issue		

Description

Strange behavior on SL5 where sometimes an exception thrown from C++ is not caught by python.
Attached a script which fails on SL5 but succeeds on SL6

History

#1 - 20 July 2015 05:00 - Rajaram, Durga

On SL5, onrec01:MAUS/.maus_trunk where it fails

```
[miceonrec01] .maus_trunk > python ./test_converter_debug.py
terminate called after throwing an instance of 'MAUS::Exception'
  what():  In branch daq_event_type
Missing required branch daq_event_type converting json->cpp at ValueItem::_SetCppChild
Aborted
[miceonrec01] .maus_trunk >
```

When processing an invalid json, an exception is thrown by

```
common_cpp/JsonCppProcessors/ObjectProcessor-inl.hh:196
```

However as seen in the script attached, the exception does not seem to propagate to python on SL5.

On SL6, a python try/catch does indeed correctly catch the throw.
Same script on onrec03:MAUS/.maus_trunk

```
[mice@miceonrec03 .maus_trunk]$ python ./test_converter_debug.py
Exception  In branch daq_event_type
Missing required branch daq_event_type converting json->cpp at ValueItem::_SetCppChild
.
----------------------------------------------------------------------
Ran 1 test in 0.004s

OK
[mice@miceonrec03 .maus_trunk]$
```

#2 - 20 July 2015 05:01 - Rajaram, Durga

- File test_converter_debug.py added

Script attached.

#3 - 20 July 2015 09:41 - Rogers, Chris

Here is a thing, I think this caused a similar problem in the past, but yours is not necessarily the same.

- Blah.so is built by linking to Exception.o, generating symbol Blah::Exception
- Bleurgh.so is built by linking to Exception.o, generating symbol Bleurgh::Exception
- If we throw from Blah.so and try to catch in Bleurgh.so; we throw a Blah::Exception and try to catch a Bleurgh::Exception. gcc RTTI thinks these are two different types and the catch fails.

Probably there is a fix using some combination of extern keyword and forward declaration, or hacking the build scripts so that MAUS::Exception is only built in libMausCpp.so, or something. I spent a morning fiddling trying to figure it out but didn't get the answer. In the end I resigned myself to catch std::exception (built in libc.so and nowhere else).

#4 - 20 July 2015 09:48 - Rogers, Chris

In which case, the patch would be

```
=== modified file 'src/py_cpp/PyConverter.cc'
--- src/py_cpp/PyConverter.cc    2015-07-10 11:43:26 +0000
+++ src/py_cpp/PyConverter.cc    2015-07-20 08:45:24 +0000
@@ -37,6 +37,10 @@
       if (data_out != NULL && py_data_out == NULL)
           throw; // memory was lost, raise it up...
       PyErr_SetString(PyExc_ValueError, (&exc)->what());
+  } catch (std::exception& exc) {
+      if (data_out != NULL && py_data_out == NULL)
+          throw; // memory was lost, raise it up...
+      PyErr_SetString(PyExc_ValueError, (&exc)->what());
   }
   return py_data_out;
 }
```

#5 - 20 July 2015 09:58 - Rogers, Chris

I pushed the patch, not sure if this will make a difference (I noticed the third_party build passed last night for example...)

#6 - 20 July 2015 12:38 - Dobbs, Adam

How much do we care about SL5 anyway? We already cannot build the unpacker properly on SL5...

Files

test_converter_debug.py	1.25 KB	20 July 2015	Rajaram, Durga