Robotics Service Bus - Bug #404
Problems when two shared libs in the same binary try to access the UnambiguousConverterMap

07/04/2011 03:05 PM - D. Klotz

Status:	Closed	**Start date:**	07/04/2011
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	C++	**Estimated time:**	0.00 hour
Target version:			

Description

There seems to be a problem when you have a binary that e.g. loads two shared libraries that both try to access the converter map, which leads to an exception "type mismatch in get for `converters': requested:rsb::converter::UnambiguousConverterMap<std::string>; actual: rsb::converter::UnambiguousConverterMap<std::string>".

In my case, the problem occurs when both the RSBAudioSender and the RSBVideoSender are loaded as shared libraries in the NaoQi process on the robot and both try to access the converter map.

Since it works when only one of the two RSB*Sender modules is loaded (regardless which one), it is probably not a problem with the typeid/-info-implementation of the older GCC/libc version on the robot, as we initially suspected. The problem could be related to the fact that the UnambiguousConverterMap is a template class and gets compiled down to both shared libraries, which leads to two versions of the same class existing in the same binary.

Unfortunately i currently don't have a simple testcase with this setup (two shared libraries using RSB and the converters in one binary), but in theory this could mean that this is not a problem that is specific to software running on Nao, but more or less a general C++ problem.

Here is the stacktrace from my original mail to Johannes and Jan, which is taken directly from the robot, so there are unfortunately quite a lot of libraries without debugging-symbols involved:

```
[New Thread 0xa70e5b90 (LWP 2347)]
properties: type mismatch in get for `converters': requested:
rsb::converter::UnambiguousConverterMap<std::string>; actual:
rsb::converter::UnambiguousConverterMap<std::string>
terminate called after throwing an instance of
'rsc::patterns::ConstructError'
  what():
boost::exception_detail::clone_impl<boost::exception_detail::error_info_injector<boost::bad_any_cast>
>: boost::bad_any_cast: failed conversion using boost::any_cast

Program received signal SIGABRT, Aborted.
[Switching to Thread 0xa70e5b90 (LWP 2347)]
0xb745346a in raise () from /lib/libc.so.6
(gdb) bt
#0  0xb745346a in raise () from /lib/libc.so.6
#1  0xb7454ae1 in abort () from /lib/libc.so.6
#2  0xb7faf8fa in __gnu_cxx::__verbose_terminate_handler() () from
/usr/lib/libstdc++.so.6
#3  0xb7fade0a in ?? () from /usr/lib/libstdc++.so.6
#4  0xb7fade39 in std::terminate() () from /usr/lib/libstdc++.so.6
#5  0xb7fadf34 in __cxa_throw () from /usr/lib/libstdc++.so.6
#6  0xaa87acf2 in rsc::patterns::Factory<std::string,
rsb::transport::OutConnector>::createInst(std::string const&,
```

rsc::runtime::Properties const&) () from
/home/nao/naoqi/lib/naoqi/librsbvideosender.so
#7 0xaa87b4c8 in rsb::Informer<AL::ALImage>::Ptr
rsb::Factory::createInformer<AL::ALImage>(rsb::Scope const&,
rsb::ParticipantConfig const&, std::string const&) () from
/home/nao/naoqi/lib/naoqi/librsbvideosender.so
#8 0xaa872629 in RsbVideoSender::sendThread() () from
/home/nao/naoqi/lib/naoqi/librsbvideosender.so
#9 0xaa872e5e in boost::detail::thread_data<boost::_bi::bind_t<void,
boost::_mfi::mf0<void, RsbVideoSender>,
boost::_bi::list1<boost::_bi::value<RsbVideoSender*> > > >::run() ()
from /home/nao/naoqi/lib/naoqi/librsbvideosender.so
#10 0xb7eeba73 in thread_proxy () from /usr/lib/libboost_thread-mt.so
#11 0xb7726c9f in start_thread () from /lib/libpthread.so.0
#12 0xb74e99be in clone () from /lib/libc.so.6

History

#1 - 04/19/2012 05:26 PM - J. Wienke

Is this still an issue? Otherwise we could simply close this until it appears again.

#2 - 04/19/2012 06:09 PM - D. Klotz

- Status changed from New to Closed

We managed to get both audio- and video-senders (and even more RSB-using modules) to be loaded as local modules (i.e. shared libraries loaded by the NaoQi process) in parallel during the Vernissage recordings, so it doesn't seem to be a problem any more. Closing this until we stumble upon it again.