

## Robotics Service Bus - Enhancement #2034

### Support participants as static members

09/30/2014 11:46 AM - J. Wienke

<b>Status:</b>	Resolved	<b>Start date:</b>	09/30/2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	J. Wienke	<b>% Done:</b>	100%
<b>Category:</b>	C++	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	rsb-0.11		
<b>Description</b>			
Make the transport implementations independent of the static destruction order so that participants can be used in static variables.			

#### Associated revisions

##### Revision 55844889 - 09/30/2014 12:58 PM - J. Wienke

Inprocess transp. independent of static destruction

This commit makes the inprocess transport independent of static destruction order so that participants using this transport can easily be used inside static variables without errors. The logic for this implementation is to couple the lifetime of the whole transport to the lifetime of participants by consequently using shared pointers for this purpose.

- src/rsb/transport/inprocess/Bus.{cpp,h}  
Remove singleton implementation to prevent static destruction without shared ptrs leaving scope. Introduce a method to retrieve a default instance that is maintained as a shared ptr instead.
- src/rsb/transport/inprocess/InConnector.{cpp,h}  
Add the Bus instance this connector is operating on as a shared ptr so that the connector keeps up the lifetime of the bus instance
- src/rsb/transport/inprocess/OutConnector.{cpp,h}  
likewise

refs #2034

##### Revision 00a7d93a - 09/30/2014 01:46 PM - J. Wienke

Socket transport independent of static destruction

This commit makes the socket transport independent of the static destruction order of C++ so that participants using this transport can easily be maintained in static member variables. This is achieved by consistently using shared pointers for managing instance lifetimes. As a consequence, all transport classes are automatically kept alive as long as they are needed, even during destruction.

- src/CMakeLists.txt  
Included new files
- src/rsb/transport/AsioServiceContext.{cpp,h}

New class to maintain the lifetime of the asio service through shared ptr destruction. Might be reusable for other transport implementations. Hence, it is added in the general transport namespace.

- src/rsb/transport/socket/Bus.{cpp,h}  
Use AsioServiceContext, add getter for this. Remove connector list which is not necessary anymore since we do not need to perform suicide. This is now done externally through shared ptr lifetime management. Made all members overridable in subclasses to prepare for implementing facade classes. No more need for shared\_from\_this.
- src/rsb/transport/socket/BusServer.{cpp,h}  
Made this something like an interface for implementing facades. Implementation moved to BusServerImpl. Adapted to accept AsioServiceContext in constructor.
- src/rsb/transport/socket/BusServerImpl.{cpp,h}  
New class with original BusServer implementation adapted to use AsioServiceContext. Also, suicide is not needed anymore since the lifetime is now managed externally.
- src/rsb/transport/socket/ConnectorBase.{cpp,h}  
This class now knows the Factory it can use to acquire Bus instances.
- src/rsb/transport/socket/Factory.{cpp,h}  
Use AsioServiceContext to maintain an asio service, Replace singleton with a getter for a default instance maintained as a shared ptr so that static destruction order is not a problem. Maintain cached Bus instances as weak pointers to prevent dependency cycles. Code refactoring to simplify searching for active, cached Bus instances. No more functions necessary to remove dying Bus instances, which would end up in cycles again.
- src/rsb/transport/socket/InConnector.{cpp,h}  
Adapted to new ConnectorBase interface
- src/rsb/transport/socket/InPullConnector.{cpp,h}  
likewise
- src/rsb/transport/socket/InPushConnector.{cpp,h}  
likewise
- src/rsb/transport/socket/LifecycledBusServer.{cpp,h}  
New class acting as a facade around BusServer instances to actively break the dependency cycle between BusServer and AsioServiceContext in case of destruction through shared pointers.
- src/rsb/transport/socket/OutConnector.{cpp,h}  
Adapted to new ConnectorBase interface
- test/rsb/transport/socket/SocketConnectorTest.cpp  
Adapted to new signatures
- test/rsb/transport/socket/SocketServerRoutingTest.cpp  
likewise

refs #2034

**Revision d53a28db - 09/30/2014 01:53 PM - J. Wienke**

Enabled participants as static members

Made transports independent of static destruction order so that

participants can now be maintained in static variables as shared ptrs.

Merge branch 'enhancement-2034'

fixes #2034

## History

---

**#1 - 09/30/2014 02:00 PM - J. Wienke**

- *Status changed from In Progress to Resolved*

- *% Done changed from 0 to 100*

Applied in changeset commit:rsb-cpp[d53a28dbac3b43a4d14f6f761937dd07421c784d.