

## Robotics Service Bus - Bug #1849

### Call tool does not complain if it can't decode a reply

04/17/2014 03:01 PM - J. Wienke

<b>Status:</b>	Feedback	<b>Start date:</b>	04/17/2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	J. Moringen	<b>% Done:</b>	30%
<b>Category:</b>	Common Lisp Tools	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	rsb-0.18		
<b>Description</b>			
The call tool simply blocks if it receives a reply for a method with a data type that cannot be deserialized. No message is given to the user.			

#### Associated revisions

##### Revision dd82cac6 - 05/11/2015 12:35 PM - J. Moringen

Fixed error-policy race condition in src/participant.lisp

refs #1849, fixes #2254

Solves the problem of connector errors not being handled by the error-policy of a newly created participant by eliminating the two causes of the problem:

1. ``make-participant-using-class'` method for ``error-hook-mixin'` connected the supplied error-policy to the error hook too late (i.e. after connectors had already been attached).
2. ``make-participant-using-class'` method for ``client'` added connectors to the configurator before installing the error-policy in the configurator.
  - src/mixins.lisp (header): updated copyright  
(`make-participant-using-class class error-hook-mixin scope`): removed  
:around qualifier; allows actual :around methods to execute code after the error-policy has been installed
  - src/participant.lisp  
(`make-participant-using-class :around class client scope`): create and set up transports, configurator and error-policy  
(`make-participant-using-class class client scope`): removed; no longer needed
  - src/event-processing/configurator.lisp (header): updated copyright  
(`shared-initialize :after configurator t`): handle processor creation properly
  - test/protocol.lisp (test `hooks-root::make-participant-hook/smoke`): adapted expected initargs
  - test/participant.lisp (header): updated copyright  
(test `participant-root::error-policy-race`): new test; make sure error policy is in effect when connectors may signal errors

## Revision 88649c71 - 05/11/2015 12:50 PM - J. Moringen

Backport: Fixed error-policy race condition in src/participant.lisp

refs #1849, refs #2254

Solves the problem of connector errors not being handled by the error-policy of a newly created participant by eliminating the two causes of the problem:

1. ``make-participant-using-class'` method for ``error-hook-mixin'` connected the supplied error-policy to the error hook too late (i.e. after connectors had already been attached).
2. ``make-participant-using-class'` method for ``client'` added connectors to the configurator before installing the error-policy in the configurator.
  - src/mixins.lisp (header): updated copyright  
(`make-participant-using-class class error-hook-mixin scope`): removed  
:around qualifier; allows actual :around methods to execute code after the error-policy has been installed
  - src/participant.lisp  
(`make-participant-using-class :around class client scope`): create and set up transports, configurator and error-policy  
(`make-participant-using-class class client scope`): removed; no longer needed
  - src/event-processing/configurator.lisp (header): updated copyright  
(`shared-initialize :after configurator t`): handle processor creation properly
  - test/protocol.lisp (test `hooks-root::make-participant-hook/smoke`): adapted expected initargs
  - test/participant.lisp (header): updated copyright  
(test `participant-root::error-policy-race`): new test; make sure error policy is in effect when connectors may signal errors

(cherry picked from commit dd82cac6a2759ae49485b7579afdd52b2c63af14)

## History

---

### #1 - 12/17/2014 03:10 PM - J. Moringen

- Target version changed from *rsb-0.11* to *rsb-0.12*

### #2 - 05/13/2015 12:03 PM - J. Moringen

- Status changed from *New* to *In Progress*

- % Done changed from 0 to 30

### #3 - 08/19/2015 04:15 PM - J. Wienke

- Target version changed from *rsb-0.12* to *rsb-0.13*

### #4 - 03/02/2016 08:55 AM - J. Moringen

- Target version changed from *rsb-0.13* to *rsb-0.14*

**#5 - 06/08/2016 09:01 PM - J. Moringen**

- *Target version changed from rsb-0.14 to rsb-0.15*

**#6 - 11/07/2016 10:32 AM - J. Moringen**

- *Target version changed from rsb-0.15 to rsb-0.16*

**#7 - 01/10/2017 12:01 PM - J. Moringen**

- *Status changed from In Progress to Feedback*

I just tried to reproduce this with the socket and spread transports and it seems to work as expected.

Can you provide a recipe for reproducing this?

**#8 - 04/10/2017 08:59 PM - J. Moringen**

- *Target version changed from rsb-0.16 to rsb-0.17*

**#9 - 10/10/2017 06:21 PM - J. Moringen**

- *Target version changed from rsb-0.17 to rsb-0.18*