

# Robotics Service Bus - Bug #2254

## Unclear fatal error raised by the logger

05/08/2015 05:38 PM - N. Köster

<b>Status:</b>	Resolved	<b>Start date:</b>	05/08/2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	J. Moringen	<b>% Done:</b>	100%
<b>Category:</b>	Common Lisp Tools	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	rsb-0.12		

### Description

#### Issue

When trying to use the logger, it crashes with no clear indication of what went wrong:

```
>>> [nkoester @ hcc] [/vol/csra/releases/trusty/lsp-csra-nightly/bin/] tools0.12 logger --style monitor --on-error=continue spread:/
200 events in queue with capacity 200.
;
; compilation unit aborted
; caught 1 fatal ERROR condition
```

```
>>> [nkoester @ hcc] [/vol/csra/releases/trusty/lsp-csra-nightly/bin/] tools0.12 logger --on-error=continue spread:/
200 events in queue with capacity 200.
;
; compilation unit aborted
; caught 1 fatal ERROR condition
```

```
>>> [nkoester @ hcc] [/vol/csra/releases/trusty/lsp-csra-nightly/bin/] tools0.12 logger
200 events in queue with capacity 200.
;
; compilation unit aborted
; caught 1 fatal ERROR condition
```

#### Version

```
>>> [nkoester @ hcc] [/vol/csra/releases/trusty/lsp-csra-nightly/bin/] tools0.12 logger --version
tools0.12 logger version 0.12.61-g7c1d36e
SBCL version      1.2.6
RSB version       0.12.42-g9dba47a
```

#### Environment

```
RSB_TRANSPORT_INPROCESS_ENABLED=0
RSB_TRANSPORT_SPREAD_ENABLED=1
RSB_MONITORING_CYCLE_TIME=3000
RSB_TRANSPORT_SPREAD_HOST=dsc
RSB_TRANSPORT_SOCKET_ENABLED=0
RSB_MONITOR=ON
```

RSB\_INTROSPECTION\_ENABLED=1  
RSB\_TRANSPORT\_SPREAD\_PORT=4814

#### Related issues:

Duplicated by Robotics Service Bus - Bug # 2219: CL send tool outputs warning...

Closed

04/07/2015

#### 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 - 05/08/2015 05:43 PM - N. Köster

the `rsb-loggercpp0.12` works fine btw

### #2 - 05/08/2015 05:52 PM - J. Moringen

N. Köster wrote:

| *the rsb-loggercpp0.12 works fine btw*

Expected, the implementations share absolutely no code.

I will probably have a patched binary ready for testing in 5 min.

### #3 - 05/08/2015 05:52 PM - J. Moringen

- *Status changed from New to In Progress*

- *% Done changed from 0 to 50*

**#4 - 05/08/2015 06:13 PM - J. Moringen**

- Status changed from *In Progress* to *Feedback*
- Assignee changed from *J. Moringen* to *N. Köster*

Please check whether [https://ci.cor-lab.org/job/rsb-tools-cl-trunk/label=ubuntu\\_trusty\\_64bit/](https://ci.cor-lab.org/job/rsb-tools-cl-trunk/label=ubuntu_trusty_64bit/) has the same problem (expectation: yes).

Please check whether [~jmoringe/pub/tools-bug2254](https://github.com/~jmoringe/pub/tools-bug2254) has the same problem (expectation: no).

(Took a bit longer because I had to build on moma, sorry)

**#5 - 05/11/2015 11:33 AM - N. Köster**

J. Moringen wrote:

| *Expected, the implementations share absolutely no code.*

Yup, I know ;) I just wanted to make clear that it is not an environment issue...

| *Please check whether [https://ci.cor-lab.org/job/rsb-tools-cl-trunk/label=ubuntu\\_trusty\\_64bit/](https://ci.cor-lab.org/job/rsb-tools-cl-trunk/label=ubuntu_trusty_64bit/) has the same problem (expectation: yes).*

Yes, same issue.

| *Please check whether [~jmoringe/pub/tools-bug2254](https://github.com/~jmoringe/pub/tools-bug2254) has the same problem (expectation: no).*

Works as expected!

| *(Took a bit longer because I had to build on moma, sorry)*

No problem! Thanks a lot for your quick reply! Unfortunately, I cannot change issue details as I am not listed as developer (or any other role) so you will have to do that...

**#6 - 05/11/2015 11:52 AM - N. Köster**

btw. will this be backported to the 0.12 release?

**#7 - 05/11/2015 12:02 PM - J. Moringen**

- Status changed from *Feedback* to *In Progress*
- Assignee changed from *N. Köster* to *J. Moringen*
- % Done changed from *50* to *90*

N. Köster wrote:

| *Please check whether [https://ci.cor-lab.org/job/rsb-tools-cl-trunk/label=ubuntu\\_trusty\\_64bit/](https://ci.cor-lab.org/job/rsb-tools-cl-trunk/label=ubuntu_trusty_64bit/) has the same problem (expectation: yes).*

| Yes, same issue.

OK.

| Please check whether ~jmoringe/pub/tools-bug2254 has the same problem (expectation: no).

| Works as expected!

I forgot to ask more precisely: the queue overflows but then operation continues?

| btw. will this be backported to the 0.12 release?

Yes.

#### #8 - 05/11/2015 12:03 PM - J. Moringen

| Unfortunately, I cannot change issue details as I am not listed as developer (or any other role) so you will have to do that...

You should be able to now.

#### #9 - 05/11/2015 12:14 PM - N. Köster

J. Moringen wrote:

| N. Köster wrote:

| | Please check whether ~jmoringe/pub/tools-bug2254 has the same problem (expectation: no).

| | Works as expected!

| I forgot to ask more precisely: the queue overflows but then operation continues?

Yes, works as expected:

- Using this with unchanged max-queued-events (aka default: 200) leads to no overflow with continuing operation.
- Using this with reduced max-queued-events (tried max-queued-events=1 to force it) leads to overflows and errors with continued operation.

Btw: I forgot to mention that the original executable works if the max-queued-events is increased (>700 seems to do the job)

#### #10 - 05/11/2015 12:34 PM - J. Moringen

N. Köster wrote:

| Yes, works as expected: - Using this with unchanged max-queued-events (aka default: 200) leads to no overflow with continuing operation.

| - Using this with reduced max-queued-events (tried max-queued-events=1 to force it) leads to overflows and errors with continued operation.

| Btw: I forgot to mention that the original executable works if the max-queued-events is increased (>700 seems to do the job)

OK, this is all as expected (assuming the --on-error continue option has been used).

I will push the fix.

**#11 - 05/11/2015 12:48 PM - J. Moringen**

- *Status changed from In Progress to Resolved*
- *% Done changed from 90 to 100*

Applied in changeset commit:rsb-cl|dd82cac6a2759ae49485b7579afdd52b2c63af14.

**#12 - 04/17/2016 07:23 PM - J. Moringen**

- *Duplicated by Bug #2219: CL send tool outputs warning messages for normal and successful calls added*