

Robotics Service Bus - Bug #344

C++ Spread: Transport-level errors are not handled properly

06/05/2011 05:37 PM - J. Moringen

Status:	New	Start date:	06/05/2011
Priority:	High	Due date:	
Assignee:		% Done:	20%
Category:	Spread Connector	Estimated time:	0.00 hour
Target version:	rsb-0.18		
Description			
The ReceiverTask class has (at least) two potential causes of errors			
- Spread errors (currently ignored)			
- Converter errors (currently not handled, ignored at call site)			
Related issues:			
Related to Robotics Service Bus - Bug # 796: Crash in ReceiverTask w/o Stackt...		Rejected	01/06/2012
Related to Robotics Service Bus - Bug # 1153: unknown data types cause crash ...		Rejected	08/29/2012
Blocked by Robotics Service Bus - Tasks # 1035: Implement error handling subs...		New	06/25/2012

Associated revisions

Revision 3814e06a - 11/28/2014 02:51 PM - J. Wienke

Terminate processing in case of receive error

Prevent 100% CPU load loops in ReceiverTask by stopping the task at the first spread error. We do not recover at all.

This is more or less just a temporary workaround for a still missing real solution for task #344 and #1035

refs #344

refs #1035

Reviewed-by: Jan Moringen <jmoringe@techfak.uni-bielefeld.de>

Revision ed871fb2 - 11/28/2014 02:51 PM - J. Wienke

Use ErrorStrategy to handle receive errors

Use the new setErrorStrategy method in the InPushConnector interface to obtain a user-specified wish on what to do in case of a receiver error.

refs #344

refs #1035

Reviewed-by: Jan Moringen <jmoringe@techfak.uni-bielefeld.de>

Revision 7bd5b422 - 11/28/2014 02:52 PM - J. Wienke

Add setErrorStrategy on InPushConnectors and call it

As a temporary solution for asynchronous receiving errors, use the ParticipantConfig::ErrorStrategy. This can be specified by the user via the configuration.

refs #344

refs #1035

Reviewed-by: Jan Moringen <jmoringe@techfak.uni-bielefeld.de>

History

#1 - 11/18/2011 03:03 PM - S. Wrede

- Target version set to 0.6

#2 - 02/27/2012 03:21 PM - J. Wienke

- Target version deleted (0.6)

postponed as it is not realistic to implement right now

#3 - 06/25/2012 10:30 PM - J. Moringen

- Target version set to rsb-0.9

#4 - 04/10/2013 02:14 AM - J. Moringen

- Subject changed from Transport-level Errors are not handled properly to Transport-level errors are not handled properly

#5 - 04/17/2013 04:14 PM - J. Moringen

- Target version changed from rsb-0.9 to rsb-0.10

#6 - 09/05/2013 05:51 PM - J. Wienke

- Subject changed from Transport-level errors are not handled properly to C++ Spread: Transport-level errors are not handled properly

- Category changed from C++ to Spread Connector

Seems to be specific to the spread transport.

#7 - 12/10/2013 11:47 PM - J. Moringen

- Target version changed from rsb-0.10 to rsb-0.11

#8 - 11/27/2014 01:22 PM - S. Wrede

- Priority changed from Normal to High

#9 - 12/17/2014 11:11 AM - J. Wienke

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

- % Done changed from 0 to 20

At least we now prevent endless 100% CPU loops. The remaining issues will have to wait for the real implementation of an error handling subsystem, which will not make it into 0.11.

#10 - 04/27/2015 11:14 AM - J. Wienke

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

#11 - 03/02/2016 08:55 AM - J. Moringen

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

#12 - 04/17/2016 07:20 PM - J. Moringen

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

#13 - 09/28/2016 02:29 PM - J. Moringen

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

#14 - 04/10/2017 08:58 PM - J. Moringen

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

#15 - 10/10/2017 06:21 PM - J. Moringen

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