Robotics Service Bus - Bug #2386
C++ spread plugin crashes when using multiple threads + informers on Windows

09/10/2015 05:14 PM - M. Goerlich

Status:	Resolved		**Start date:**	09/10/2015
Priority:	Normal		**Due date:**	
Assignee:	J. Moringen		**% Done:**	100%
Category:	Spread Connector		**Estimated time:**	0.00 hour
Target version:	rsb-0.17			

Description

The spread library does not seem to be thread safe under windows (though stated within the documentation).

This touches RSB since it also causes crashes when one creates two rsb informers and sends data through two different threads. These infomers share the same spread connection.

See attached code, funny windows dump thingy and funky stacktrace-screenshot for more info.

It looks like a possible fix might be to protect the send and receive calls with a mutex-lock.

Associated revisions

Revision fb774a0e - 05/19/2017 01:25 PM - J. Moringen

Fixed(?) Spread thread-safety issue in src/rsb/transport/spread/SpreadConnection.{h,cpp}

At least for sending messages, the Spread client library does not seem
to be thread-safe on win32 (despite what the documentation says).

This fix seems to work but has only been verified experimentally. In
particular, it is not clear, whether all other combinations of Spread
client API functions are thread-safe.

fixes #2386

- src/rsb/transport/spread/SpreadConnection.{h,cpp} (header): updated
  copyright
  (SpreadConnection::mutex): new member; for critical section in send
  on win32
  (SpreadConnection::send): turned Spread client API calls into
  critical section on win32

Revision e933fbb5 - 05/19/2017 01:26 PM - J. Moringen

Backport: Fixed(?) Spread thread-safety issue in src/rsb/transport/spread/SpreadConnection.{h,cpp}

At least for sending messages, the Spread client library does not seem
to be thread-safe on win32 (despite what the documentation says).

This fix seems to work but has only been verified experimentally. In
particular, it is not clear, whether all other combinations of Spread
client API functions are thread-safe.

refs #2386

- src/rsb/transport/spread/SpreadConnection.{h,cpp} (header): updated
  copyright
  (SpreadConnection::mutex): new member; for critical section in send
  on win32
  (SpreadConnection::send): turned Spread client API calls into
  critical section on win32

(cherry picked from commit fb774a0e93a4f9947285f320f294b7e92600652b)

History

#1 - 09/14/2015 11:07 AM - J. Wienke
- Status changed from New to In Progress
- Assignee set to J. Wienke
- Target version set to rsb-0.13

This exception is raise by our code as a result of a failure return code from the spread library. Unfortunately. In your screenshot I cannot see the string
contained in the exception. Is there any way for you to reproduce this and to print out the what() of the thrown exception?

#2 - 09/14/2015 12:12 PM - M. Goerlich

Hey Johannes,

probably it would be best if I showed you the bug on the windows machine in the CSRA environment. I will be there tomorrow starting from at the
latest 10 o'clock, open end. The same goes for Wednesday and Thursday.

#3 - 09/24/2015 03:17 AM - J. Moringen
- Subject changed from c++ spread plugin crashes when using multiple threads + informers to C++ spread plugin crashes when using multiple threads
+ informers on Windows

#4 - 03/03/2016 12:11 PM - 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 - 04/10/2017 08:59 PM - J. Moringen
- Target version changed from rsb-0.16 to rsb-0.17

#8 - 05/19/2017 01:16 PM - J. Moringen
- Assignee changed from J. Wienke to J. Moringen

#9 - 05/19/2017 01:30 PM - J. Moringen
- Status changed from In Progress to Resolved
- % Done changed from 0 to 100

Applied in changeset commit:rsb-spread-cpp|fb774a0e93a4f9947285f320f294b7e92600652b.

Files

thread-test.cpp	3.86 KB	09/10/2015	M. Goerlich
stacktrace.png	154 KB	09/10/2015	M. Goerlich
thread-test.dmp.tar.gz	20.5 MB	09/10/2015	M. Goerlich