

# Robotics Service Bus - Bug #140

## rsbtest\_spread randomly crashes

11/10/2010 07:19 PM - J. Wienke

<b>Status:</b>	Resolved	<b>Start date:</b>	11/10/2010
<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>			
<b>Description</b>			
Threading issue with stopping the executor, I suppose:			
<pre>(gdb) bt #0 0x000000002039420 in ?? () #1 0x00007f3948b38ae3 in rsb::spread::ReceiverTask::execute (this=0x20339d0, t=0x2039090) at /homes/jwienke/nao/RSB/src/rsb/transport/spread/ReceiverTask.cpp:93 #2 0x00007f3948b31ad0 in boost::_mfi::mf1&lt;void, rsb::spread::ReceiverTask, rsb::util::Task&lt;void&gt;*&gt;::operator() (this=0x20390a0, p=0x20339d0, a1=0x2039090) at /usr/include/boost/bind/mem_fn_template.hpp:162 #3 0x00007f3948b31878 in boost::_bi::list2&lt;boost::_bi::value&lt;rsb::spread::ReceiverTask*&gt;, boost::arg&lt;1&gt; &gt;::operator()&lt;boost::_mfi::mf1&lt;void, rsb::spread::ReceiverTask, rsb::util::Task&lt;void&gt;*&gt;, boost::_bi::list1&lt;rsb::util::Task&lt;void&gt;*&gt;*&gt; &gt; (   this=0x20390b0, f=..., a=...) at /usr/include/boost/bind/bind.hpp:306 #4 0x00007f3948b31768 in boost::_bi::bind_t&lt;void, boost::_mfi::mf1&lt;void, rsb::spread::ReceiverTask, rsb::util::Task&lt;void&gt;*&gt;, boost::_bi::list2&lt;boost::_bi::value&lt;rsb::spread::ReceiverTask*&gt;, boost::arg&lt;1&gt; &gt; &gt;::operator()&lt;rsb::util::Task&lt;void&gt;*&gt; (this=0x20390a0, a1=@0x7f3946b8fb20) at /usr/include/boost/bind/bind_template.hpp:32 #5 0x00007f3948b316a3 in boost::detail::function::void_function_obj_invoker1&lt;boost::_bi::bind_t&lt;void, boost::_mfi::mf1&lt;void, rsb::spread::ReceiverTask, rsb::util::Task&lt;void&gt;*&gt;, boost::_bi::list2&lt;boost::_bi::value&lt;rsb::spread::ReceiverTask*&gt;, boost::arg&lt;1&gt; &gt; &gt;, void, rsb::util::Task&lt;void&gt;*&gt;::invoke (function_obj_ptr=..., a0=0x2039090) at /usr/include/boost/function/function_template.hpp:153 #6 0x000000000466b88 in boost::function1&lt;void, rsb::util::Task&lt;void&gt;*&gt;::operator() (this=0x2039098, a0=0x2039090) at /usr/include/boost/function/function_template.hpp:1013 #7 0x00000000046502f in rsb::util::Task&lt;void&gt;::run (this=0x2039090) at /homes/jwienke/nao/RSB/test/./src/rsb/util/Task.h:75 #8 0x00000000046af05 in boost::_mfi::mf0&lt;void, rsb::util::Task&lt;void&gt; &gt;::operator() (this=0x2039818, p=0x2039090) at /usr/include/boost/bind/mem_fn_template.hpp:49 #9 0x00000000046ae74 in boost::_bi::list1&lt;boost::_bi::value&lt;rsb::util::Task&lt;void&gt;*&gt; &gt;::operator()&lt;boost::_mfi::mf0&lt;void, rsb::util::Task&lt;void&gt; &gt;, boost::_bi::list0&gt; (this=0x2039828, f=..., a=...) at /usr/include/boost/bind/bind.hpp:246 #10 0x00000000046ae21 in boost::_bi::bind_t&lt;void, boost::_mfi::mf0&lt;void, rsb::util::Task&lt;void&gt; &gt;, boost::_bi::list1&lt;boost::_bi::value&lt;rsb::util::Task&lt;void&gt;*&gt; &gt; &gt;::operator() (this=0x2039818) at /usr/include/boost/bind/bind_template.hpp:20 #11 0x0000000004699c2 in boost::detail::thread_data&lt;boost::_bi::bind_t&lt;void, boost::_mfi::mf0&lt;void, rsb::util::Task&lt;void&gt; &gt;, boost::_bi::list1&lt;boost::_bi::value&lt;rsb::util::Task&lt;void&gt;*&gt; &gt; &gt; &gt;::run (this=0x2039710)   at /usr/include/boost/thread/detail/thread.hpp:56 #12 0x00007f394883ab70 in thread_proxy (param=&lt;value optimized out&gt;) at libs/thread/src/pthread/thread.cpp:130 #13 0x00007f39481139ca in start_thread (arg=&lt;value optimized out&gt;) at pthread_create.c:300 #14 0x00007f39476c270d in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:112 #15 0x0000000000000000 in ?? ()</pre>			

### Associated revisions

Revision 53632c8c - 01/05/2011 08:38 PM - J. Wienke

Eventhough the symptoms were different after the Task changes in RSC, a similar bug like in #140 existed in the current code. The handler of the informer task was called from the port after the task was destructed. Now properly wait on all events from the informer task in a separate waiter object and only shutdown if we are sure that no more events are sent.

fixes #140

## History

---

### #1 - 01/05/2011 08:40 PM - J. Wienke

- Status changed from New to Resolved
- Assignee changed from S. Wrede to J. Wienke

### #2 - 01/05/2011 08:40 PM - J. Wienke

- % Done changed from 0 to 100