

# Robotics Systems Commons - Enhancement #2063

## simplify clumsy usage of DebugTools

10/18/2014 10:49 AM - R. Haschke

<b>Status:</b>	In Progress	<b>Start date:</b>	10/18/2014
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	90%
<b>Category:</b>	Debug Tools	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	rsb-1.0		
<b>Description</b>			
Using backtrace functions requires a DebugTools object. Why did you choose for this rather clumsy design?			
Instead of:			
<pre>rsc::debug::DebugToolsPtr dbg = rsc::debug::DebugTools::newInstance(); cout &lt;&lt; dbg-&gt;formatBacktrace(dbg-&gt;createBacktrace());</pre>			
I suggest the following function-based usage:			
<pre>cout &lt;&lt; rsc::debug::formatBacktrace(rsc::debug::createBacktrace());</pre>			
Having an overloaded convenience function, this can further simplify to:			
<pre>cout &lt;&lt; rsc::debug::formatBacktrace();</pre>			
As there are no data elements, there is no need for an DebugTools object, isn't it?			
<b>Related issues:</b>			
Blocks Robotics Service Bus - Feature # 2051: Qt-based graphical event logger		<b>In Progress</b>	<b>10/09/2014</b>

### History

#### #1 - 10/22/2014 10:20 PM - R. Haschke

- Status changed from Feedback to Resolved
- Assignee changed from J. Wienke to R. Haschke
- % Done changed from 0 to 90

- extract + demangle C++ function names on Linux commit:43ee009
- removed class DebugTools, expose functions createBacktrace() + formatBacktrace() directly commit:38c6e40
- added Windows implementation for createBacktrace(): please verify on Windows machine! commit:78798d1

source branch: enhancement-2063

#### #2 - 10/23/2014 03:07 PM - J. Wienke

- Category set to Debug Tools
- Status changed from Resolved to In Progress

This issue cannot be resolved as long as things aren't in the master branch. ;)

Thanks for the work. I am fine with the general API changes, however, there are several formal issues that need to be resolved before we can integrate this into master.

1. Your feature branch is based on the URI feature branch. It needs to be rebased on master without the URI changes. If you rename it to

enhancement-2063, it will also automatically the jenkins merge simulator jobs so that you can get feedback from jenkins on your changes (assuming there is no new bug in jenkins for this feature).

2. Please do not use inline functions defined in the headers. These end up being compiled in the client programs and not in the library. Hence, internal changes to RSC will require a recompilation of client code, which effectively increases coupling and breaks separation of concerns.

3. Please document under which assumptions the demangle functions works (e.g. compiler). Any chance to get the code more readable? E.g. by using real c++ strings.

4. I am not sure whether we can simply include the StackWalker as it is to not violate license agreements. We need to find this out.

5. In any case, the windows code needs to adhere to our coding standards. In all 3 windows-related files, tabs and windows line ends are used, our license headers are missing and indentation and line break rules are off.

6. Please either integrate the required features of MyStackWalker directly into the original StackWalker and limit its functionality to what we actually require or give this class a more descriptive name to explain the purpose why it exists. Adapting the StackWalker to the functionality we need, might also be sufficient to move it further apart from the original code base so that we declare this as our own code under our license. I am not sure about this.

Apart from the general licensing issue, would you adapt your branch accordingly?

### #3 - 10/23/2014 05:04 PM - R. Haschke

I can work on that some time.

I suggest to keep the StackWalker code as is. I didn't touched anything there yet and I don't want to.

### #4 - 10/24/2014 10:26 AM - J. Wienke

Robert Haschke wrote:

*I can work on that some time.  
I suggest to keep the StackWalker code as is. I didn't touched anything there yet and I don't want to.*

So, I'd propose to move the windows stracktrace support to a new issue and feature branch so that we can continue with the API changes without having to discuss the license issues. Moreover, this also feels more like a separate feature.

### #5 - 10/24/2014 11:07 AM - R. Haschke

I worked on all the points you pointed out. However, I didn't touched StackWalker itself.

You should decide how to deal with the BSD license of that code or extract the necessary functionality. Actually StackWalker is much more powerful than required in rsc context: It can create stacktraces of different threads or even processes.

### #6 - 02/11/2015 01:54 AM - R. Haschke

- Assignee changed from R. Haschke to J. Wienke

### #7 - 02/11/2015 02:10 AM - R. Haschke

- Blocks Feature #2051: Qt-based graphical event logger added

### #8 - 02/24/2015 10:33 AM - S. Wrede

- Target version set to rsb-0.12

### #9 - 04/27/2015 11:16 AM - J. Wienke

- Target version changed from rsb-0.12 to rsb-1.0

#10 - 06/30/2016 12:38 PM - J. Wienke

- Assignee deleted (J. Wienke)