Robotics Service Bus - Bug #1245
Singleton factories do not work as expected on windows

11/19/2012 11:29 AM - J. Wienke

Status:	Resolved	**Start date:**	11/19/2012
Priority:	Normal	**Due date:**	
Assignee:	J. Wienke	**% Done:**	100%
Category:	C++	**Estimated time:**	0.00 hour
Target version:	rsb-0.9		

Description

As described in #1231, multiple instances of the singletons occur across build units on windows leading to unexpected effects.
I would propose the following strategy to solve the problem:

- in 0.8:
  - retain the singletons but mark there use as deprecated
  - provide getters for singleton instances which are definitely created in the RSB dll
- in 0.9 or later:
  - Remove the singleton base class of the respective factories so that they can only be used using the getters with predictable behavior on all platforms.

Any objections to this procedure?

Related issues:

Related to Robotics Service Bus - Enhancement # 1247: Remove singleton backwa...	**Resolved**	**11/19/2012**
Blocked by Robotics Service Bus - Bug # 1231: Connector and EventProcessingSt...	**Resolved**	**11/07/2012**

Associated revisions

Revision 27624814 - 11/19/2012 06:43 PM - J. Wienke

Prevent the presence of multiple singleton factories on windows.

Due to the separate memory management of windows for different DLLs the existing singleton implementation does not work there. All uses of the singleton were replace by appropriate getters to a specific instance.
In detail:

- Singleton base classes were made private to reuse the instantiation support in static initializer code of the existing singleton implementation
- Appropriate getter methods were added falling back to the now private singleton implementation
- getInstance methods were redeclared using custom code going back to the getters. These methods are now marked deprecated and singleton use is discouraged.

For EventSendingStrategyFactory and the receiving counterpart these changes could not be performed because they use a specific factory implementation from RSC. However, they are probably never used from the outside.

refs #1245

Revision 295b9bb7 - 11/19/2012 10:47 PM - J. Wienke

Prevent the presence of multiple singleton factories on windows.

Due to the separate memory management of windows for different DLLs the existing singleton implementation does not work there. All uses of the singleton were replace by appropriate getters to a specific instance.
In detail:

- Singleton base classes were made private to reuse the instantiation support in static initializer code of the existing singleton implementation
- Appropriate getter methods were added falling back to the now private singleton implementation
- getInstance methods were redeclared using custom code going back to the getters. These methods are now marked deprecated and singleton use is discouraged.

For EventSendingStrategyFactory and the receiving counterpart these changes could not be performed because they use a specific factory implementation from RSC. However, they are probably never used from the outside.

refs #1245

Revision 3e304f85 - 11/19/2012 10:49 PM - J. Wienke

Provide getters for factories instead of using singletons to overcome windows memory management issues.

Backwards compatibility is preserved (from an API point of view) but should be removed in one of the next versions. These functions are marked deprecated.

fixes #1245

Merge branch 'bug-1245'

History

#1 - 11/19/2012 12:50 PM - J. Moringen

It may be possible to use the rsc::patterns::Singleton as a private base class instead of removing it entirely. This way, clients would not be able to call rsc::patterns::Singleton::getInstance directly and code duplication could still be avoided.

#2 - 11/19/2012 07:02 PM - J. Wienke

- Status changed from New to In Progress
- Assignee set to J. Wienke
- % Done changed from 0 to 70

Jan, I have committed a fix for this issue to the respective bug branch, commit 27624814. Can you have a quick look at this as it is a rather larger change?

#3 - 11/19/2012 09:49 PM - J. Moringen

Looks OK, but rsb::transport::ConnectorFactory<>::getInstance() could use a small explanatory comment.

#4 - 11/19/2012 10:54 PM - J. Wienke

- Status changed from In Progress to Resolved

- % Done changed from 70 to 100

Applied in changeset rsb-cpp|commit:3e304f85f06b317efb7cc9998b5d8c9c3a74cee8.