Robotics Service Bus - Bug #1231
Connector and EventProcessingStrategie factory singletons are instantiated multiple times on windows

11/07/2012 07:19 AM - V. Losing

Status:	Resolved	**Start date:**	11/07/2012
Priority:	High	**Due date:**	
Assignee:	J. Wienke	**% Done:**	100%
Category:	C++	**Estimated time:**	0.00 hour
Target version:	rsb-0.7		

Description

When for example rsb_listener.exe is used in combination with spread, the listener crashes immediately(when rsb.conf is configured properly).

this is the stack from VS 2010:

msvcr100d.dll!_NMSG_WRITE(int rterrnum) Line 217 C
msvcr100d.dll!abort() Line 61 + 0x7 bytes C
msvcr100d.dll!terminate() Line 115 C++
rsb_listener.exe!__CxxUnhandledExceptionFilter(_EXCEPTION_POINTERS * pPtrs) Line 70 C++
kernel32.dll!7685003f()
[Frames below may be incorrect and/or missing, no symbols loaded for kernel32.dll]
msvcr100d.dll!_getptd_noexit() Line 500 C
00000011()
ntdll.dll!778274df()
ntdll.dll!778273bc()
ntdll.dll!77827261()
ntdll.dll!7780b459()
ntdll.dll!7780b42b()
ntdll.dll!7780b3ce()
ntdll.dll!777c0133()
KernelBase.dll!7603b9bc()
msvcr100d.dll!_free_dbg(void * pUserData, int nBlockUse) Line 1267 + 0xc bytes C++
msvcr100d.dll!_unlock(int locknum) Line 375 C
msvcr100d.dll!_heap_alloc_dbg_impl(unsigned int nSize, int nBlockUse, const char * szFileName, int nLine, int * errno_tmp) Line 507 + 0x7 bytes C++
msvcr100d.dll!_heap_alloc_dbg_impl(unsigned int nSize, int nBlockUse, const char * szFileName, int nLine, int * errno_tmp) Line 504 + 0xc bytes C++
msvcr100d.dll!_nh_malloc_dbg_impl(unsigned int nSize, int nhFlag, int nBlockUse, const char * szFileName, int nLine, int * errno_tmp) Line 239 + 0x19 bytes C++
00c5f250()
msvcr100d.dll!_CxxThrowException(void * pExceptionObject, const _s__ThrowInfo * pThrowInfo) Line 157 C++
> rsbcore.dll!rsc::patterns::Factory<std::basic_string<char,std::char_traits<char>,std::allocator<char> >,rsb::transport::InPushConnector>::createInst(const std::basic_string<char,std::char_traits<char>,std::allocator<char> > & key, const rsc::runtime::Properties & properties_) Line 361 + 0x105 bytes C++
rsbcore.dll!rsb::Factory::createInPushConnectors(const rsb::ParticipantConfig & config) Line 299 + 0x46 bytes C++
rsbcore.dll!rsb::Factory::createListener(const rsb::Scope & scope, const rsb::ParticipantConfig & config) Line 210 + 0x34 bytes C++
rsb_listener.exe!main(int argc, char * * argv) Line 55 + 0x60 bytes C++
rsb_listener.exe!__tmainCRTStartup() Line 555 + 0x19 bytes C
rsb_listener.exe!mainCRTStartup() Line 371 C
kernel32.dll!7681339a()

ntdll.dll!777e9ef2()
ntdll.dll!777e9ec5()

Related issues:

Blocks Robotics Service Bus - Bug # 1245: Singleton factories do not work as ...	**Resolved**	**11/19/2012**

Associated revisions

Revision f38d61c9 - 11/09/2012 02:38 PM - J. Moringen

Removed unused getOutFactoryInstance() in src/rsb/Factory.{h,cpp}

refs #1231

This function was originally used to prevent
rsb::transport::OutFactory::getInstance() from being called from the
template member function rsb::Factory::createInformer<T>() which lead
to Problems with Windows memory management. However, due to some
refactoring of rsb::Factory, rsb::transport::OutFactory::getInstance()
is now called from rsb::Factory::createOutConnectors() which is not a
template member function. Hence getOutFactoryInstance() should no
longer be needed.

- src/rsb/Factory.{h,cpp}: removed unused function
  getOutFactoryInstance()

Revision 37c574dd - 11/15/2012 01:54 PM - J. Wienke

Prevent multiple instantiations of Transport factories on windows due to DLL memory management and template singleton classes.

refs #1231

Signed-off-by: Johannes Wienke <jwienke@techfak.uni-bielefeld.de>

Revision 1cd4fbca - 11/19/2012 02:03 PM - J. Wienke

Fix bug 1231 by preventing the direct use of template singletons factories outside the rsb main library (dll on windows).

The previous behavior resulted in empty connector and strategy lists in the rsb info tool because of mutiple instantiations of the "singleton" due to the win

Merge branch 'bug-1231'

refs #1231

Revision da08e919 - 11/19/2012 02:04 PM - J. Wienke

backport: Fix bug 1231 by preventing the direct use of template singletons factories outside the rsb main library (dll on windows).

The previous behavior resulted in empty connector and strategy lists in the rsb info tool because of mutiple instantiations of the "singleton" due to the win

refs #1231

History

#1 - 11/07/2012 10:00 AM - J. Moringen

- Description updated

- Category set to C++

- Target version set to rsb-0.7

#2 - 11/07/2012 10:02 AM - J. Moringen

Did you create a rsb.conf file? If so, can you post its contents?

Is it possible, that you built RSB without Spread support? This controlled by a CMake option called BUILD_SPREAD_TRANSPORT.

#3 - 11/07/2012 11:25 AM - V. Losing

- File rsb.conf added

I've checked the option for spread support in the cmake cache, and it is checked.
added my rsb.conf

#4 - 11/07/2012 01:45 PM - J. Wienke

I would have suspected something like this:
http://stackoverflow.com/questions/2322095/why-does-this-program-crash-passing-of-stdstring-between-dlls

#5 - 11/07/2012 04:00 PM - J. Moringen

@Johannes: I don't think this is only an access violation. The final part of the backtrace may show some heap-related problem, though.

However, the root cause seems to be that the Factory<InPushConnector> does not find a requested implementation (that's why i asked for rsb.conf and BUILD_SPREAD_TRANSPORT). Didn't you add a Windows-specific workaround for the problem that multiple instances of singletons could exist? Maybe this is the same problem? Or maybe our recent changes in rsc::patterns::Singleton caused this. What do you think?
@V. Losing: just to be sure, could you

1. check BUILD_SOCKET_TRANSPORT
2. try an empty rsb.conf
3. try the following rsb.conf

```
[transport.spread]
enabled = 1
```

and report the respective results. Thanks, and sorry for the trouble.

#6 - 11/07/2012 05:28 PM - V. Losing

1)BUILD_SOCKET_TRANSPORT is checked
2)with empty rsb.conf nothing happens, but i'm not sure how to send and receive data.
I started the rsb_informer.exe but received nothing with the rsb_listener.exe
3) same problem with the suggested rsb.conf

#7 - 11/08/2012 11:11 AM - J. Moringen

Thanks for trying that. Can you also attach the output of

1. rsb_version --verbose
2. rsb_listener with

```
[rsc.logging]
rsc.LEVEL = TRACE
rsb.LEVEL = TRACE
```

in rsb.conf

#8 - 11/08/2012 02:19 PM - V. Losing

- File rsb_version_output.txt added
- File rsb_listener_output.txt added

here are the demanded outputs...

#9 - 11/08/2012 05:31 PM - J. Moringen

Thanks, I think, I see the problem now (excerpt from attachment:rsb_version_output.txt)

```
Connectors
ConnectorFactory<class rsb::transport::InPullConnector>[
]
ConnectorFactory<class rsb::transport::InPushConnector>[
]
ConnectorFactory<class rsb::transport::OutConnector>[
]
```

@Johannes: Looks like the singleton problem, right?

#10 - 11/09/2012 12:37 AM - J. Wienke

Oh yes. So the trick was that the singleton instance needs to be created inside the rsb DLL. For this purpose the dll then needs to provide a getter on that instance. The reason is, that every dll and the main binary have distinct memory management on windows and hence no instance is found in the main binary code if it was previously created in the dll and vice versa.

#11 - 11/09/2012 02:28 PM - J. Moringen

Strangely, the getOutFactoryInstance() hack is not currently in effect. rsb-cpp compiles happily with the function removed.

However, the rsb::transport::{InPush,InPull,Out}Factory::getInstance() calls should end up in rsb.dll since they are performed in non-template methods rsb::Factory::create{InPush,InPull,Out}Connectors in rsb/Factory.cpp.

The *registration* is in a different *compilation unit* (src/transport/transports.cpp), but should still be in rsb.dll. Could this be a problem?

#12 - 11/13/2012 09:44 PM - J. Moringen

- Assignee set to J. Wienke

#13 - 11/15/2012 05:24 PM - J. Wienke

While trying to reproduce your issue I ended up being unable to reproduce it again on our build infrastructure. Eventhough there are some problems, e.g. with displaying the available connectors in rsb_version, I cannot reproduce the core crash here. Could it be that your RSB installation did not find spread and is hence built without the spread transport?
Can you do the following things please:

1. From the patched version we prepared yesterday, please give me the output of rsb_version again. There should be some connectors visible
2. Attach the CMakeCache.txt file from the build folder.
3. Change your config to enable the socket transport and disable the spread transport and try launching the listener again. Does it still crash?
4. Repeat the same, but disable all transports and enable transport.inprocess. Does it crash now?

#14 - 11/16/2012 08:22 AM - V. Losing

- File rsb-version.txt added
- File CMakeCache.txt added

I've added the requested outputs.

3. with socket transport it doesn't crash, I'm not sure how to test a send and listen chain. How does rsb_send.exe work? rsb_send.exe [scope] [filename]. I've tried "rsb_send.exe / rsb.conf" and got some output by the send cmd-line, but nothing happended on the listener terminal.

4. no crash here...

#15 - 11/16/2012 09:24 AM - J. Wienke

Ok, I suspect this is the real problem (apart from the other ones we started fixing):

```
SPREAD_INCLUDE_DIRS:PATH=SPREAD_INCLUDE_DIRS-NOTFOUND
```

I suspect that the spread transport actually wasn't compiled, which is also in line with the rsb-version output:

```
Connectors
ConnectorFactory<class rsb::transport::InPullConnector>[
  ConnectorInfo[inprocess, schemas = {inprocess}, remote = 0, options = {enabled}]
  ConnectorInfo[socket, schemas = {socket}, remote = 1, options = {host, port, server, tcpnodelay, enabled}]
```

During the initial CMake phase, CMake searches for spread (executable, libraries and include paths) and if all of them are found, the spread transport will be built, otherwise not. In that case a message should appear in the CMake output.

What you need to do is let CMake find the required paths. Internally we use this find macro:
https://code.cor-lab.org/projects/rsc/repository/revisions/master/entry/cmake/Modules/FindSpread.cmake
It assumes the following folder layout to be found under SPREAD_ROOT:

```
sbin/ (or bin)
  spread.exe
lib/ (or bin)
  libspread*.lib
include/
  sp.h etc.
```

Please ensure that your installation of spread looks according to this scheme and point cmake the the folder containing this hierarchy by setting the SPREAD_ROOT variable during configuration time. Afterwards, ensure from the output that the spread transport will really be built. If this succeeds, apart from the display problems of rsb-version, everything should already work.

#16 - 11/16/2012 01:40 PM - V. Losing

I've added the include dir from spread, and rebuilded rsb, now the exception is gone :D.
I will test receive and send by a simple project at the weekend to make sure everything runs correctly.
Thanks a lot!

#17 - 11/19/2012 11:26 AM - J. Wienke

- Subject changed from RSB crashes on Win7 to Connector and EventProcessingStrategie factory singletons are instantiated multiple times on windows

I will remap this bug to the actual minor problem of viewing the factories to prevent a renaming of the branch for fixes.

The issue is that in tools like version/info the factories for connectors and event processing strategies are requested manually using the singleton template class. This results in different instantiations on windows inside the rsb dll and the binaries like e.g. info and version and hence no connectors and strategies are visible. However, they are still usable from inside the rsb dll.

The overall rsb::Factory is not affected at all, as it is only instantiated inside the client. We might hit another problem here if multiple deployment units are used by the client and hence end up with different instances of the factory, but I will move this to a new issue.

#18 - 11/19/2012 02:11 PM - J. Wienke

- Status changed from New to Resolved
- % Done changed from 0 to 100

Changes of the wip branch are now in 0.7 and master.

Files

rsb.conf	68 Bytes	11/07/2012	V. Losing

rsb_version_output.txt	2.57 KB	11/08/2012	V. Losing
rsb_listener_output.txt	1.97 KB	11/08/2012	V. Losing
rsb-version.txt	3.17 KB	11/16/2012	V. Losing
CMakeCache.txt	23.4 KB	11/16/2012	V. Losing