

Robotics Service Bus - Bug #1230

Unintuitive error message upon misconfiguration

11/06/2012 06:35 PM - S. Wrede

Status:	Closed	Start date:	11/06/2012
Priority:	Normal	Due date:	
Assignee:	J. Moringen	% Done:	0%
Category:	C++	Estimated time:	0.00 hour
Target version:	rsb-0.9		

Description

Using the current master of everything, I just experienced the following error which was a bit puzzling due to a -- let's say -- minimal error message.

The configuration was as follows:

- No Spread plugin installed
- In the configuration Spread transport was **disabled**
- Test case was a CCA binary without any special configuration

```
Sebastians-MacBook-Air:build swrede$ ./pingpong
```

```
1352222613963 cca [INFO]: Ping Node component constructed.
```

```
1352222613964 rsb.Factory [INFO]: Registering default implementations
```

```
1352222613964 rsb.converter.Repository [INFO]: Registering converter *rsb::converter::BoolConverter[wireType = std::string, wireSchema = bool, dataType = bool] at 0x7fb80b40e810
```

```
1352222613964 rsb.converter.Repository [INFO]: Registering converter *rsb::converter::ByteArrayConverter[wireType = std::string, wireSchema = .*, dataType = bytearray] at 0x7fb80b40ed00
```

```
1352222613964 rsb.converter.Repository [INFO]: Registering converter *EventIdConverter[wireType = std::string, wireSchema = .rsb.protocol.EventId, dataType = rsb::EventId] at 0x7fb80b40ede0
```

```
1352222613964 rsb.converter.Repository [INFO]: Registering converter *rsb::converter::StringConverter[wireType = std::string, wireSchema = utf-8-string, dataType = std::string] at 0x7fb80b40ef40
```

```
1352222613964 rsb.converter.Repository [INFO]: Registering converter *rsb::converter::Uint64Converter[wireType = std::string, wireSchema = uint64, dataType = unsigned long long] at 0x7fb80b40eff0
```

```
1352222613964 rsb.converter.Repository [INFO]: Registering converter *rsb::converter::VoidConverter[wireType = std::string, wireSchema = void, dataType = void] at 0x7fb80b40f330
```

```
1352222613964 rsb.transport.ConnectorFactory<rsb::transport::InPushConnector> [INFO]: Registering connector inprocess for schemas {inprocess}
```

```
1352222613964 rsb.transport.ConnectorFactory<rsb::transport::InPushConnector> [INFO]: Registering connector socket for schemas {socket}
```

```
1352222613964 rsb.transport.ConnectorFactory<rsb::transport::InPullConnector> [INFO]: Registering connector inprocess for schemas {inprocess}
```

```
1352222613964 rsb.transport.ConnectorFactory<rsb::transport::InPullConnector> [INFO]: Registering connector socket for schemas {socket}
```

```
1352222613964 rsb.transport.ConnectorFactory<rsb::transport::OutConnector> [INFO]: Registering connector inprocess for schemas {inprocess}
```

```
1352222613964 rsb.transport.ConnectorFactory<rsb::transport::OutConnector> [INFO]: Registering connector socket for schemas {socket}
```

```
1352222613964 rsb.Factory [INFO]: Processing plugin configuration
```

```
1352222613965 rsc.plugins.Configurator [DEBUG]: Processing option #(transport, inprocess, enabled) = `1'
```

```
1352222613965 rsc.plugins.Configurator [DEBUG]: Processing option #(transport, socket, enabled) = `1'
```

```
1352222613965 rsc.plugins.Configurator [DEBUG]: Processing option #(transport, spread, enabled) = `0'
```

```
1352222613965 rsc.plugins.Configurator [INFO]: No searchpath configured; using default: #("/Users/swrede/.rsb0.8/plugins", "/usr/local/Cellar/rsb/HEAD/lib/rsb0.8/plugins")
```

```
1352222613965 rsc.plugins.Manager [INFO]: Adding path "/Users/swrede/.rsb0.8/plugins"
```

```
1352222613965 rsc.plugins.Manager [INFO]: Ignoring non-existent path `"/Users/swrede/.rsb0.8/plugins"'
```

```
1352222613965 rsc.plugins.Manager [INFO]: Adding path "/usr/local/Cellar/rsb/HEAD/lib/rsb0.8/plugins"
1352222613965 rsc.plugins.Manager [INFO]: Ignoring non-existent path "/usr/local/Cellar/rsb/HEAD/lib/rsb0.8/plugins"
1352222613965 rsb.ParticipantConfig [DEBUG]: Skipping disabled transport Transport[name = inprocess, converters = {},
options = p{ enabled: "0" }]
1352222613965 rsb.ParticipantConfig [DEBUG]: Skipping disabled transport Transport[name = inprocess, converters = {},
options = p{ enabled: "0" }]
1352222613965 rsb.ParticipantConfig [DEBUG]: Skipping disabled transport Transport[name = spread, converters = {}, options
= p{ enabled: "0" }]
1352222613965 rsb.Factory [DEBUG]: Default config rsb::ParticipantConfig[qosSpec = QualityOfServiceSpec[ordering = 10,
reliability = 20], errorStrategy = 0, transports = {Transport[name = socket, converters = {}, options = p{ enabled: "1" }],
Transport[name = spread, converters = {}, options = p{ enabled: "0" }], Transport[name = inprocess, converters = {}, options =
p{ enabled: "1" }]}, eventReceivingStrategy = rsb::ParticipantConfig::EventProcessingStrategy[name = parallel, options = p{ }],
eventSendingStrategy = rsb::ParticipantConfig::EventProcessingStrategy[name = direct, options = p{ }], options = p{ }]]
```

The sole output of the exception handler (printing `ex->what()` to the console):

```
spread
```

It turned out that when I commented out the Spread section from the configuration (although it was disabled) the error went away.

In my opinion:

- the message in the exception is not informative for naive users
- we need to decide what the default behavior in that situation shall be.

Related issues:

Related to Robotics Service Bus - Tasks # 1513: Document exception handling i...

Resolved

05/27/2013

History

#1 - 11/09/2012 09:37 AM - D. Klotz

This is a slightly different case (RSB not linked with the spread library, but spread is the only enabled transport), but I think the basic problem (no useful error message in case of a non-working transport configuration) is the same:

```
david@camaris:~$ rsb_listener /
1352450033202 rsb.ParticipantConfig [DEBUG]: Skipping disabled transport Transport[name = inprocess, converters = {}, options = p{
enabled: "0" }]
1352450033202 rsb.ParticipantConfig [DEBUG]: Skipping disabled transport Transport[name = socket, converters = {}, options = p{ enabled:
"0" }]
1352450033202 rsb.Factory [DEBUG]: Trying to add connector Transport[name = spread, converters = {}, options = p{ enabled: "1" }]
1352450033202 rsb.Factory [DEBUG]: Supplied connector options p{ enabled: "1" }
1352450033202 rsb.Factory [DEBUG]: Converter configuration for transport `spread': {}
1352450033202 rsb.Factory [DEBUG]: Selected converters for transport `spread': *UnambiguousConverterMap[converters = {.*:
*rsb::converter::ByteArrayConverter[wireType = std::string, wireSchema = .*, dataType = bytearray] at 0x7fe5f1c099e0
.rsb.protocol.EventId: *EventIdConverter[wireType = std::string, wireSchema = dummy, dataType = rsb::EventId] at 0x7fe5f1c09ac0
bool: *rsb::converter::BoolConverter[wireType = std::string, wireSchema = bool, dataType = bool] at 0x7fe5f1c094f0
uint64: *rsb::converter::UInt64Converter[wireType = std::string, wireSchema = uint64, dataType = unsigned long long] at 0x7fe5f1c09d10
utf-8-string: *rsb::converter::StringConverter[wireType = std::string, wireSchema = utf-8-string, dataType = std::string] at 0x7fe5f1c09c20
void: *rsb::converter::VoidConverter[wireType = std::string, wireSchema = void, dataType = void] at 0x7fe5f1c09dc0] at 0x7fe5f1c05ab0
libc++abi.dylib: terminate called throwing an exception
Abort trap: 6
```

#2 - 04/04/2013 06:00 PM - J. Moringen

- *Subject changed from Unintuitive Error Message Upon Misconfiguration to Unintuitive error message upon misconfiguration*
- *Status changed from New to Feedback*
- *Assignee set to J. Moringen*

I think, this issue is fixed in current master:

1. If the configuration refers to (but does not try to enable) non-existent transports (Sebastian's original report), no exception is raised.
2. If an attempt is made to enable a non-existent transport (David's report), an exception with a message similar to
no implementation of interface `rsb::transport::InPushConnector' found for specified key `spread'
is raised.

This may still result in the behavior reported by David, but the rsb_listener program (now called example-listener) and/or MacOS are to blame for not reporting the exception properly.

@David, @Sebastian: Please confirm or deny.

#3 - 05/27/2013 04:55 PM - J. Moringen

- *Status changed from Feedback to Closed*

The current behavior is OK in all cases Johannes and I could test on MacOS and Linux (But we can still document this, see #1513).

Reopen, if problems remain.