Robotics Systems Types - Enhancement #1520
I have to manually resolve ambiguity for .rst.geometry.Pose and .rst.vision.Image

05/29/2013 11:18 AM - Anonymous

Status:	Resolved	**Start date:**	05/29/2013
Priority:	High	**Due date:**	
Assignee:	J. Moringen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	rsb-0.9		

Description

When working with debian package librstconverters-amarsi0.9 I have to manually resolve the ambiguity for .rst.geometry.Pose and .rst.vision.Image (only those two).

Excerpt from the converter registry:

```
...
1369818603231 rsb.converter.repository [INFO]: Registering converter *rsb::converter::ProtocolBufferConverter<rst::geometry::Pose>[wireType = std::string, wireSchema = .rst.geometry.Pose, dataType = rst::geometry::Pose] at 0x10036e0
...
1369818603231 rsb.converter.repository [INFO]: Registering converter *rst::converters::boost::TransformationPoseConverter[wireType = std::string, wireSchema = .rst.geometry.Pose, dataType = boost::multi_array<double, 2ul, std::allocator<double> >] at 0x1004580
1369818603231 rsb.converter.repository [INFO]: Registering converter *rst::converters::boost::TwoDPoseConverter[wireType = std::string, wireSchema = .rst.geometry.Pose, dataType = boost::multi_array<double, 1ul, std::allocator<double> >] at 0x1004b40
terminate called after throwing an instance of 'std::runtime_error'
  what():  Ambiguous converter set for wire-type `std::string' and wire-schema `.rst.geometry.Pose': candidate data-types are {boost::multi_array<double, 2ul, std::allocator<double> >, rst::geometry::Pose, boost::multi_array<double, 1ul, std::allocator<double> >}; hint: add a configuration option `transport.<name>.converter.cpp.".rst.geometry.Pose" = <one of {boost::multi_array<double, 2ul, std::allocator<double> >, rst::geometry::Pose, boost::multi_array<double, 1ul, std::allocator<double> >}>' to resolve the ambiguity.
Aborted (core dumped)
```

So as for Pose, there are three converters for the wire-schema .rst.geometry.Pose.

1. First of all: Is this erroneous behavior / configuration?
2. If this is intended behavior, can be somehow change the default behavior? It doesn't seem very convenient, that you have to manually resolve ambiguity before working with the rst converters.

Associated revisions

Revision 8d8d11d8 - 06/05/2013 05:45 PM - J. Moringen

Put converter registration into separate plugins

fixes #1520

The problem was that loading the rstrsbconvertersstable plugin tried
to register converters for all stable RST types in addition to the
alvision, OpenCV, boost and RCI converters (if those features were

enabled). This lead to ambiguous converter setups when creating RSB listeners.

- cpp/src/CMakeLists.txt: generate additional plugins for alvision, OpenCV, boost and RCI

Revision ad3a4bbb - 06/05/2013 05:45 PM - J. Moringen

Put converter registration into separate plugins

fixes #1520

The problem was that loading the rstrsbconvertersstable plugin tried to register converters for all stable RST types in addition to the alvision, OpenCV, boost and RCI converters (if those features were enabled). This lead to ambiguous converter setups when creating RSB listeners.

- cpp/src/CMakeLists.txt: generate additional plugins for alvision, OpenCV, boost and RCI

Revision c15100f1 - 07/26/2016 01:16 PM - J. Moringen

Put converter registration into separate plugins

fixes #1520

The problem was that loading the rstrsbconvertersstable plugin tried to register converters for all stable RST types in addition to the alvision, OpenCV, boost and RCI converters (if those features were enabled). This lead to ambiguous converter setups when creating RSB listeners.

- cpp/src/CMakeLists.txt: generate additional plugins for alvision, OpenCV, boost and RCI

History

#1 - 05/29/2013 11:38 AM - Anonymous

So this seems to be intended behavior and hard to avoid.

However, what is especially inconvenient with this behavior is, that I have to manually configure in order to be able to use rst-converters **and** I have to do this configuration for data-types I potentially don`t even want to use, don`t care about and maybe don`t even know / understand (if I only want to use different converters).

This problem will become even more pressing with more and more converters being added to rst (which is what we want, I guess).

#2 - 06/01/2013 06:14 PM - J. Moringen

- Tracker changed from Bug to Enhancement

- Status changed from Feedback to In Progress

- Assignee set to J. Moringen

Arne and I considered separating rst-converters into more plugins such that loading individual plugins never leads to ambiguities.

#3 - 06/05/2013 06:10 PM - J. Moringen

- Status changed from In Progress to Resolved

- % Done changed from 0 to 100

Applied in changeset rst-converters|commit:8d8d11d8385f940643505a4bcfc53ecbb3b09180.