

Robotics Service Bus - Bug #1408

Converter registration should be possible without rewriting default participant configuration

02/11/2013 03:34 PM - D. Klotz

Status:	Resolved	Start date:	02/11/2013
Priority:	Normal	Due date:	
Assignee:	J. Wienke	% Done:	100%
Category:	Python	Estimated time:	0.00 hour
Target version:	rsb-0.9		

Description

I currently have the problem that a library I am using in my code (XTT) is registering some converters for the datatypes it uses. Since in Python this is currently only possible by rewriting the default participant config (the docs recommend calling `rsb.setDefaultParticipantConfig(rsb.ParticipantConfig.fromDefaultSources())` after registering the converters), this is what XTT is doing internally (albeit still with the old `rsb.__defaultParticipantConfig = ...` syntax).

Now in my code (e.g. in some unit tests), I explicitly set a default participant config, but this gets completely overwritten as soon as I call XTTs initialization code. And since this is the only documented way of registering converters, I don't really blame XTT for doing it like that.

In the end, registering new converters in Python should be possible without completely rewriting the default configuration, i.e. by appending to it (or doing whatever the other languages are doing for this purpose).

Associated revisions

Revision 1877000f - 05/28/2013 10:19 AM - J. Wienke

Implement lazy converter map preparation.

So far, inside the participant configuration, the converter map was prepared once initially with the first access to the participant config and afterwards loaded from a cached variable. As a consequence it was impossible to add a converter to the global converter map without regenerating the whole `ParticipantConfig`. This, however, had the drawback that previous manual modifications to the configuration were erased. With this change the converter map for a new participant is created in a lazy fashion each time a converter map for a participant is requested. Hence, new converters can now be added to the global converter map and are automatically available for any new participant.

- `rsb/__init__.py`: Implemented lazy loading of converter map
- `test/coretest.py`: Unit test added to verify the behavior

refs #1408

Revision d98723b3 - 05/28/2013 10:34 AM - J. Wienke

Update example for converter registration.

Calling `setDefaultParticipantConfig` is not required anymore.

refs #1408

Revision 24849541 - 05/28/2013 10:36 AM - J. Wienke

Update python converter registration.

With the lazy way of creating converter maps for individual participants it is not required anymore to manually recreate the global participant configuration.

- Remove instructions to recreate the configuration
- Add a note to explicitly warn about this behavior.

refs #1408

Revision 8ec3b1e6 - 05/28/2013 02:55 PM - J. Wienke

Enable converter registration without changes to ParticipantConfig

Make the creation of ConverterMap instances dynamic so that new converters can be added to the global map all the time without requiring a newly generated default participant configuration.

fixes #1408

Merge branch 'bug-1408'

Revision f0258e71 - 05/28/2013 03:05 PM - J. Wienke

Update python converter registration.

With the lazy way of creating converter maps for individual participants it is not required anymore to manually recreate the global participant configuration.

- Remove instructions to recreate the configuration
- Add a note to explicitly warn about this behavior.

refs #1408

Revision bbb228f7 - 05/28/2013 03:06 PM - J. Wienke

Adapt manual to python converter registration changes

Reflect the lazy creation of converter maps for individual participants.

- Warn about previous usage

refs #1408

Merge branch 'bug-1408'

History

#1 - 02/20/2013 05:46 PM - D. Klotz

Are there any ideas on how and when this could be changed? This makes it really hard to write rsb-using python libraries, since they can only register the converters by replacing the participant config (and thereby overwriting any changes to it that e.g. the library-using client-program might have done before that).

Maybe someone could shed some light on why this is done differently in Python than e.g. in C++, where AFAIK you can register a new converter without replacing the default participant config?

#2 - 05/07/2013 01:55 PM - J. Wienke

- Target version changed from *rsb-0.7* to *rsb-0.9*

We should not forget to fix this in 0.9. It's a severe restriction.

#3 - 05/27/2013 06:36 PM - J. Wienke

- Assignee changed from *J. Moringen* to *J. Wienke*

#4 - 05/27/2013 09:40 PM - J. Wienke

- Status changed from *New* to *In Progress*

#5 - 05/28/2013 11:23 AM - J. Wienke

- % Done changed from *0* to *80*

I have pushed respective changes to the implementation and the manual. Can someone please review them and verify the intended behavior?

#6 - 05/28/2013 03:08 PM - J. Wienke

- Status changed from *In Progress* to *Resolved*

- % Done changed from *80* to *100*

Applied in changeset `rsb-python|commit:8ec3b1e64674144c7f1165338d82ce35dbf6d447`.