Robotics Service Bus - Bug #1404
It's impossible to add a 'void' method to a local RPC server in Python

02/07/2013 04:08 PM - D. Klotz

Status:	Resolved	**Start date:**	02/07/2013
Priority:	Normal	**Due date:**	
Assignee:	J. Wienke	**% Done:**	100%
Category:	Python	**Estimated time:**	0.00 hour
Target version:	rsb-0.9		

Description

It is impossible to register a method that does not return a value for RPC servers created in Python, because there does not seem to be a counterpart of the VoidType used for this purpose in other languages.

The addMethod call on a (local) server object in Python has a default value of 'object' for the replyType parameter. This creates a problem because by default, there is no converter available for type 'object'.

This leads to the problem that when you try to register a method like in the following example, the server throws a (not very helpful non-descriptive "KeyError") exception when the registered method is called by a client:

```
[...]
def say(sentence):
    print("Saying: '%s'" % sentence)

if __name__ == '__main__':
    server = rsb.createServer("/foo/bar")
    server.addMethod("say", say, str)
[...]
```

In the end, there should be way to register a method that does not return a value. why not use pythons NoneType for this? And IMHO this type should then be the default for the replyType parameter of server.addMethod, or it should not have a default parameter value at all (since the unusable default of 'object' is quite misleading).

Associated revisions

Revision 8f92140d - 02/08/2013 12:42 PM - J. Wienke

Backport: Make converter selection really dynamic in case of Server methods.

- The reply event was attached with the declared data type for the method instead of the one of the data type.
- Added a unit test for this case

refs #1404

Revision 748c0753 - 02/08/2013 12:42 PM - J. Wienke

Make converter selection really dynamic in case of Server methods.

- The reply event was attached with the declared data type for the method instead of the one of the data type.
- Added a unit test for this case

refs #1404

History

#1 - 02/07/2013 04:10 PM - D. Klotz

- Description updated

#2 - 02/07/2013 08:44 PM - J. Moringen

- Status changed from New to Feedback

Well, there are None and type(None):

```
[...]
def say(sentence):
    print("Saying: '%s'" % sentence)

if __name__ == '__main__':
    server = rsb.createServer("/foo/bar")
    server.addMethod("say", say, str, type(None))
[...]
```

Does this solve your Problem?

#3 - 02/07/2013 08:45 PM - J. Moringen

Btw: could you add the target version to this issue?

#4 - 02/08/2013 10:12 AM - D. Klotz

- Target version set to rsb-0.7

server.addMethod("say", say, str, type(None)) works, server.addMethod("say", say, str, None) does not (although support for the latter syntax would IMHO be nice). Thanks for the heads up!

But if there is already support for this, why is the default value for the replyType object (i.e. not usable in the default configuration)? I still think that either there should be no default value (so you would be required to specify a type) or the default value should be something usable.

#5 - 02/08/2013 11:12 AM - D. Klotz

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

This problem is also present in the master, the default still seems to be object.

#6 - 02/08/2013 11:20 AM - D. Klotz

This is what happens on the server side, the client just doesn't get an answer:

```
dklotz@rhodenit:[~/workspace/MSS]python simple_rpc_server.py
No handlers could be found for logger "rsb.transport.rsbspread.InConnector"
Saying: 'Hello World!'
Exception in thread Thread-1:
```

```
Traceback (most recent call last):
  File "/usr/lib/python2.7/threading.py", line 551, in __bootstrap_inner
    self.run()
  File "/usr/lib/python2.7/threading.py", line 504, in run
    self.__target(*self.__args, **self.__kwargs)
  File "/homes/dklotz/local/x86_64/lib/python2.7/site-packages/rsb_python-0.9.0-py2.7.egg/rsb/util.py", line 326, in __worker
    self.__delFunc(receiver.receiver, message)
  File "/homes/dklotz/local/x86_64/lib/python2.7/site-packages/rsb_python-0.9.0-py2.7.egg/rsb/eventprocessing.py", line 135, in __deliver
    action(event)
  File "/homes/dklotz/local/x86_64/lib/python2.7/site-packages/rsb_python-0.9.0-py2.7.egg/rsb/patterns/__init__.py", line 318, in
_handleRequest
    self.informer.publishEvent(reply)
  File "/homes/dklotz/local/x86_64/lib/python2.7/site-packages/rsb_python-0.9.0-py2.7.egg/rsb/__init__.py", line 1260, in publishEvent
    self.__configurator.handle(event)
  File "/homes/dklotz/local/x86_64/lib/python2.7/site-packages/rsb_python-0.9.0-py2.7.egg/rsb/eventprocessing.py", line 357, in handle
    self.__sendingStrategy.handle(event)
  File "/homes/dklotz/local/x86_64/lib/python2.7/site-packages/rsb_python-0.9.0-py2.7.egg/rsb/eventprocessing.py", line 201, in handle
    connector.handle(event)
  File "/homes/dklotz/local/x86_64/lib/python2.7/site-packages/rsb_python-0.9.0-py2.7.egg/rsb/transport/rsbspread/__init__.py", line 393, in
handle
    converter = self.getConverterForDataType(event.type)
  File "/homes/dklotz/local/x86_64/lib/python2.7/site-packages/rsb_python-0.9.0-py2.7.egg/rsb/transport/__init__.py", line 177, in
getConverterForDataType
    return self.__converterMap.getConverterForDataType(dataType)
  File "/homes/dklotz/local/x86_64/lib/python2.7/site-packages/rsb_python-0.9.0-py2.7.egg/rsb/converter.py", line 145, in
getConverterForDataType
    raise KeyError, dataType
KeyError: <type 'object'>
```

#7 - 02/08/2013 11:49 AM - D. Klotz

- Assignee changed from J. Moringen to J. Wienke

#8 - 02/08/2013 12:43 PM - J. Wienke

- Status changed from Feedback to Resolved

- % Done changed from 0 to 100