Using subscriber/callback function inside of a class C++

3

C++

For the life of me I cannot seem to figure out why this code is not working and throws up a lot of errors:

asked Dec 9 '13
Phorce
57 ●10 ●11 ●16

updated Jan 29 '14
ngrennan
1 ●1 ●1

```
class Example {

public:

    Example(ros::NodeHandle n )
    {
        sub = n.subscribe("/camera/depth_registered/points", 1000, &this->callBack);

    }

    void callBack(const sensor_msgs::PointCloud2Ptr& msg)
    {


    }

protected:

    ros::Subscriber sub;
};
```

I have initialised the `ros::Subscriber sub` inside of the class members but cannot seem to figure out why it's giving me an error.

Any ideas?

EDIT:

Here are the error messages:

```
error: no matching function for call to ‘ros::NodeHandle::subscribe(const char [32], int,
<unresolved overloaded function type>)’ /src/talker.cpp:11:71: note: candidates are:
/opt/ros/hydro/include/ros/node_handle.h:379:14: note: template<class M, class T>
ros::Subscriber ros::NodeHandle::subscribe(const string&, uint32_t, void (T::*)(M), T*, const
ros::TransportHints&)
```

Comments

What exact errors? When are you getting them? During compile or when running? Initial guess: Try adding a this to the subscribe call an specify the classname instead of this-> for the subscribe.
dornhege (Dec 9 '13)

add a comment

Question Tools

Follow

1 follower

subscribe to rss feed

Stats

Asked:	Dec 9 '13
Seen:	6,904 times
Last updated:	Dec 09 '13

Related questions

- Turtlebot doesn't move in a straight line
- odometry messages to C++ node silently dropped
- service error
- C++ , FCL: Compilations Issues with Simple Collision Geometry
- how to set the start direction of SPHERO to the x-axis
- catkin_make started failing to generate .h from .msg
- About listener.cpp tutorial. disobey cpp grammar?
- import rospy when calling a python script from a c++ executable
- Service client C++ tutorial simple question
- Problems with rqt in C++ [Solved] [closed]

1 Answer

Sort by » oldest newest most voted

22

Callbacks with class member functions are a little tricky. "Regular" functions are merely a pointer to some code in memory. Class member functions have additional state information, namely the object instance they belong to, so you cannot just plug a member function into a regular function pointer and expect it to work.

answered Dec 10 '13
roehling
1626 ●16 ●26 ●42
http://www.fkie.fraunh...

There are two easy ways to solve this. Firstly, the ROS developers anticipated this problem and provided a neat alternative `subscribe` method that accepts member functions and the corresponding object like this:

```
sub = n.subscribe("/camera/depth_registered/points", 1000, &Example::callBack, this);
```

The `&Example::callBack` is the function pointer to the member function, and `this` is the object instance for which you want to have the callback called.

But even if they had forgotten, the second option is to use `boost::bind()`, a very powerful tool to bind arguments to arbitrary functions, which supports class member functions as well:

```
sub = n.subscribe("/camera/depth_registered/points", 1000, boost::bind(&Example::callBack, this, _1));
```

The syntax is slightly more complicated, but it is much more versatile (read the Boost documentation for details).

Comments

link

I have the same problem , I used the boost::bind,but I have new error information
the code: _sub_object = _nh.subscribe("/perception/object",1,boost::bind(&MotionCore::_callback_from_perception_obstacle,this));
the error: In file included from /home/westeast/git/enmodel/src/trajectory/nodes/final


westeast (**May 13 '17**)

add a comment

Your Answer

Please start posting anonymously - your entry will be published after you log in or create a new account.

Add Answer