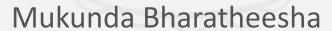
1.5.2

ROS Actions – code illustration



Creating an action file

Let's first create our action folder and our new action file.

- \$ mkdir action
- \$ cd action
- \$ nano CounterWithDelay.action

We first define the goal message as the number of counts to count up to. Then a result message which is a string message for the result and a feedback message to indicate the numbers of counts elapsed.

Creating an action file

CounterWithDelay.action

```
uint32 num_counts
---
string result_message
---
uint32 counts_elapsed
```

Now we update the CMakeLists.txt file with the vim editor in the action files subsection. (#L25)

CounterWithDelay.action (#L27)

We need to add the actionlib package that provides all the ROS action functionalities. We need to add it at all dependency locations. (#L3,37,45)

actionlib actionlib_mgs

Update the package.xml file also with the dependency. (#L21)

Now run catkin build or in short catkin b.

Action message

Don't forget to source your ROS workspace first!

\$ source \$HOME/hrwros_ws/devel/setup.bash

If we type the complete action name

\$ rosmsg show hrwros_msgs/CounterWithDelayAction

We will see the complete message definition.

We need to pay attention to where our message definitions are located as a lot of the message is for the actionlib package.

If you want to see the message definition, you can go to the hrwros_msgs package folder.

\$ roscd hrwros_msgs

Action message

And run the following script.

```
$ rosrun actionlib_msgs genaction.py -o msg
action/CounterWithDelay.action
```

There we find all the action definitions.

```
$ rosmsg show hrwros_msgs/
```