

Activity 2: Understanding ROS 2 nodes

The Ros 2 Graph

Over the next few tutorials, you will learn about a series of core ROS 2 concepts that make up what is referred to as the “ROS (2) graph”.

The ROS graph is a network of ROS 2 elements processing data together at one time. It encompasses all executables and the connections between them if you were to map them all out and visualize them.

Nodes in ROS 2

Each node in ROS should be responsible for a single, module purpose (e.g. one node for controlling wheel motors, one node for controlling a laser range-finder, etc). Each node can send and receive data to other nodes via topics, services, actions, or parameters.

Source setup environment

```
source /opt/ros/foxy/setup.bash
```

1. ros2 run

The command `ros2 run` launches an executable from a package.

```
ros2 run <package_name> <executable_name>
```

To run `turtlesim`, open a new terminal(`ctrl+shift+T`), and enter the following command:

```
ros2 run turtlesim turtlesim_node
```

The `turtlesim` window will open.

Here, the package name is `turtlesim` and the executable name is `turtlesim_node`.

We still don't know the node name, however. You can find node names by using `ros2 node list`.

2. ros2 node list

`ros2 node list` will show you the names of all running nodes. This is especially useful when you want to interact with a node, or when you have a system running many nodes and need to keep track of them.

Open a new terminal(`ctrl+shift+T`) while `turtlesim` is still running in the other one, and enter the following command:

```
ros2 node list
```

The terminal will return the node name:

```
/turtlesim
```

Open another new terminal and start the teleop node with the command:

```
ros2 run turtlesim turtle_teleop_key
```

Here, we are searching the turtlesim package again, this time for the executable named turtle_teleop_key.

Return to the terminal where you ran ros2 node list and run it again. You will now see the names of two active nodes:

```
/turtlesim  
/teleop_turtle
```

3. Remapping

Remapping allows you to reassign default node properties, like node name, topic names, service names, etc., to custom values.

Now, let's reassign the name of our /turtlesim node. In a new terminal, run the following command:

```
ros2 run turtlesim turtlesim_node --ros-args --remap __node:=my_turtle
```

Since you're calling ros2 run on turtlesim again, another turtlesim window will open. However, now if you return to the terminal where you ran `ros2 node list`, and run it again, you will see three node names:

```
/turtlesim  
/teleop_turtle  
/my_turtle
```

4. Ros2 node info

Now that you know the names of your nodes, you can access more information about them with:

```
ros2 node info <node_name>
```

To examine your latest node, `my_turtle`, run the following command:

```
ros2 node info /my_turtle
```

`ros2 node info` returns a list of subscribers, publishers, services, and actions (the ROS graph connections) that interact with that node (Don't worry, these concepts will be explained next!). The output should look like this:

```
/my_turtle
Subscribers:
  /parameter_events: rcl_interfaces/msg/ParameterEvent
  /turtle1/cmd_vel: geometry_msgs/msg/Twist
Publishers:
  /parameter_events: rcl_interfaces/msg/ParameterEvent
  /rosout: rcl_interfaces/msg/Log
  /turtle1/color_sensor: turtlesim/msg/Color
  /turtle1/pose: turtlesim/msg/Pose
Services:
  /clear: std_srvs/srv/Empty
  /kill: turtlesim/srv/Kill
  /reset: std_srvs/srv/Empty
  /spawn: turtlesim/srv/Spawn
  /turtle1/set_pen: turtlesim/srv/SetPen
  /turtle1/teleport_absolute: turtlesim/srv/TeleportAbsolute
  /turtle1/teleport_relative: turtlesim/srv/TeleportRelative
  /my_turtle/describe_parameters: rcl_interfaces/srv/DescribeParameters
  /my_turtle/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
  /my_turtle/get_parameters: rcl_interfaces/srv/GetParameters
  /my_turtle/list_parameters: rcl_interfaces/srv/ListParameters
  /my_turtle/set_parameters: rcl_interfaces/srv/SetParameters
  /my_turtle/set_parameters_atomically:
rcl_interfaces/srv/SetParametersAtomically
Action Servers:
  /turtle1/rotate_absolute: turtlesim/action/RotateAbsolute
Action Clients:
```

Now try running the same command on the `/teleop_turtle` node, and see how its connections differ from `my_turtle`.