

1부 11장_ROS2 토픽

요약

- Topic 기능 사용 방법
 - `ros2 topic list`
 - `ros2 topic info`
 - `ros2 topic echo`
 - `ros2 topic bw`
 - `ros2 topic hz`
 - `ros2 topic delay`
 - `ros2 topic pub`
 - `—once` 옵션
 - `—rate {hz}` 옵션
 - `ros2 bag`
 - `record`
 - `info`
 - `play`

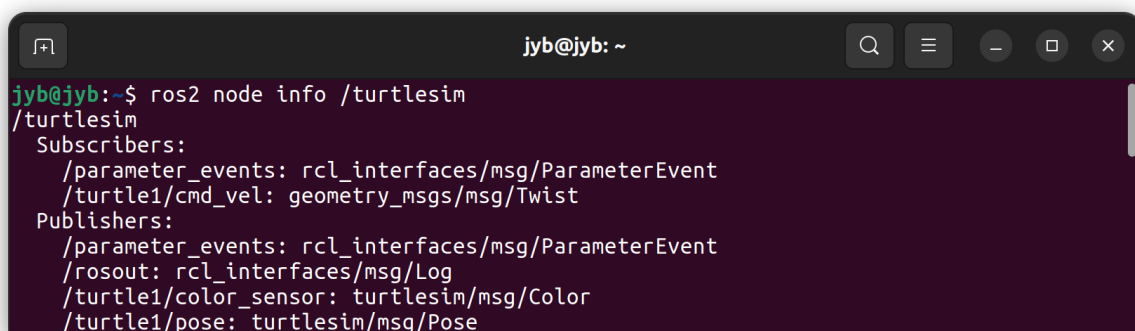
토픽 목록 확인

- `turtlesim` 노드 하나를 연다.

```
ros2 run turtlesim turtlesim_node
```

- 노드 정보를 확인한다.
 - `turtlesim` 노드는 `cmd_vel` 메시지를 서브스크라이브하고 있다.
 - `turtlesim` 노드는 `color_sensor`, `pose` 메시지를 퍼블리시하고 있다.

```
ros2 node info /turtlesim
```

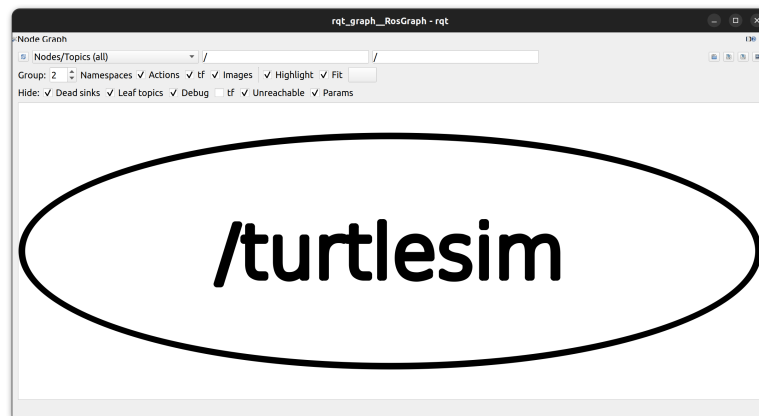


```
jyb@jyb: ~  
jyb@jyb:~$ ros2 node info /turtlesim  
/turtlesim  
Subscribers:  
  /parameter_events: rcl_interfaces/msg/ParameterEvent  
  /turtle1/cmd_vel: geometry_msgs/msg/Twist  
Publishers:  
  /parameter_events: rcl_interfaces/msg/ParameterEvent  
  /rosout: rcl_interfaces/msg/Log  
  /turtle1/color_sensor: turtlesim/msg/Color  
  /turtle1/pose: turtlesim/msg/Pose
```

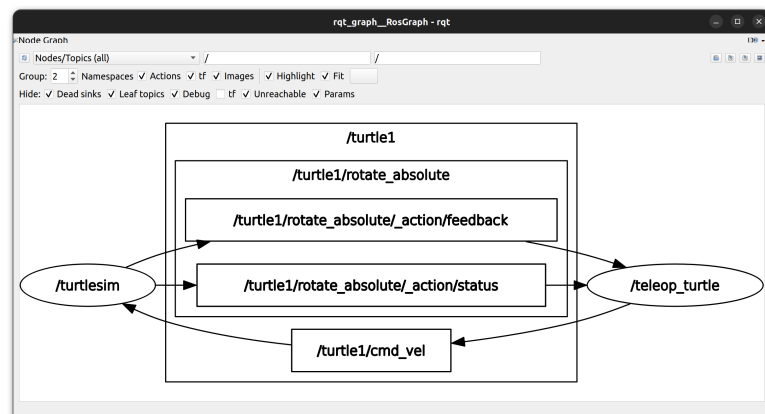
- 현재 개발환경에서 동작 중인 모든 노드들의 토픽 정보를 확인한다.
 - 현재는 turtlesim 노드만 실행되었기 때문에 turtlesim 노드가 퍼블리시, 서브스크라이브하는 메시지만 표시되고 있다.
 - -t 옵션: 각 메시지의 type도 함께 표시한다.

```
ros2 topic list -t
```

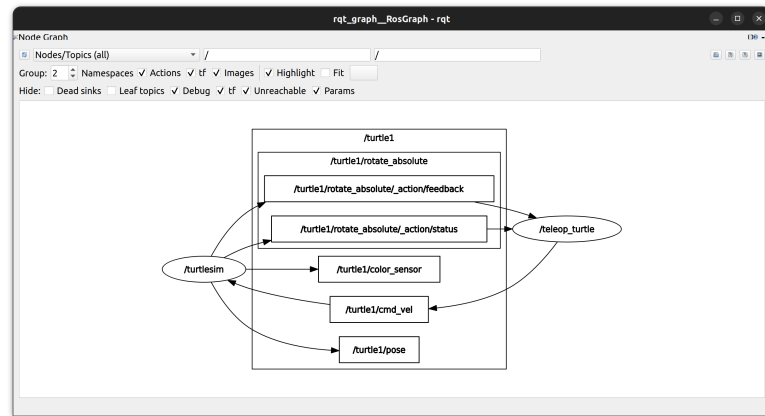
```
jyb@jyb: ~
jyb@jyb:~$ ros2 topic list -t
/parameter_events [rcl_interfaces/msg/ParameterEvent]
/rosout [rcl_interfaces/msg/Log]
/turtle1/cmd_vel [geometry_msgs/msg/Twist]
/turtle1/color_sensor [turtlesim/msg/Color]
/turtle1/pose [turtlesim/msg/Pose]
jyb@jyb:~$
```



- turtle_teleop_key 노드를 추가했을 때 변화
 - topic list는 변화가 없다.
 - rqt_graph
 - 기본 default 창



- Dead sinks, Leaf topics 해제하여 숨겨져 있던 color_sensor, pose 토픽도 확인할 수 있다.



토픽 정보 확인

```
ros2 topic info /turtle1/cmd_vel
```

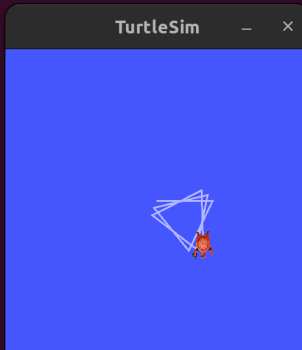
```
jyb@jyb: ~
jyb@jyb:~$ ros2 topic info /turtle1/cmd_vel
Type: geometry_msgs/msg/Twist
Publisher count: 1
Subscription count: 1
jyb@jyb:~$
```

토픽 내용 확인

- “ros2 topic echo” 명령으로 메시지 내용을 실시간으로 확인할 수 있다.
- 명령을 보낸 후 teleop_key로 이동하면 현재 turtle의 업데이트된 상태가 표시된다.
- 모든 메시지 단위는 SI를 기본으로 사용한다.

```
ros2 topic echo /turtle1/cmd_vel
```

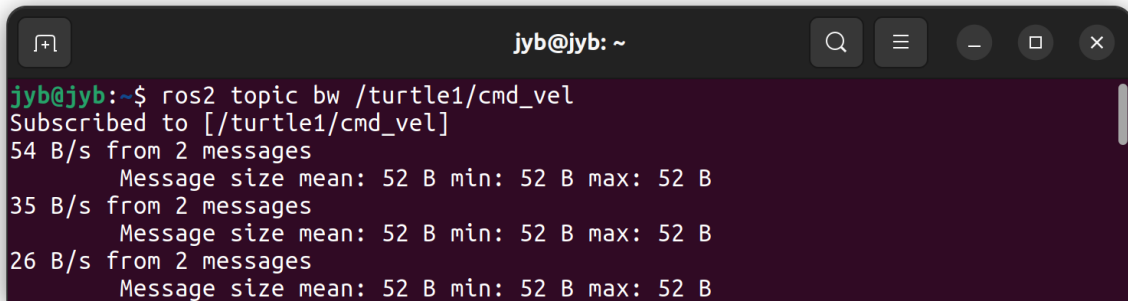
```
jyb@jyb: ~
jyb@jyb:~$ ros2 topic echo /turtle1/cmd_vel
linear:
  x: 0.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: -2.0
---
linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
```



토픽 대역폭 확인

- “ros2 topic bw” 명령으로 메시지의 대역폭, 즉 송수신 받는 토픽 메시지의 크기 (초당 대역폭)를 확인할 수 있다.
- 사용하는 메시지 형태 및 주기에 따라 대역폭이 달라질 수 있다.

```
ros2 topic bw /turtle1/cmd_vel
```

A terminal window titled 'jyb@jyb: ~' showing the command 'ros2 topic bw /turtle1/cmd_vel'. The output shows three lines of data: '54 B/s from 2 messages', '35 B/s from 2 messages', and '26 B/s from 2 messages'. Each line is followed by 'Message size mean: 52 B min: 52 B max: 52 B'.

토픽 주기 확인

- “ros2 topic hz” 명령으로 메시지의 전송 주기를 확인할 수 있다.
- turtle_teleop_key 노드에서 얼마나 자주 cmd_vel 토픽을 퍼블리시 하는 지 확인할 수 있다.

```
ros2 topic hz /turtle1/cmd_vel
```

A terminal window titled 'jyb@jyb: ~' showing the command 'ros2 topic hz /turtle1/cmd_vel'. The output shows two lines of data: 'average rate: 0.703' and 'average rate: 0.701'. Each line is followed by 'min: 1.351s max: 1.496s std dev: 0.07267s window: 2' and 'min: 1.351s max: 1.496s std dev: 0.05947s window: 3'. The terminal ends with '^Cjyb@jyb:~\$ ^C'.

토픽 지연 시간 확인

- 토픽은 RMW 및 네트워크 장비를 거치기 때문에 latency, 즉 지연 시간이 존재한다.
- 메시지 내에 header stamp 메시지를 사용하고 있다면 ros2 topic delay 명령어를 이용해 지연 시간을 확인할 수 있다.
 - https://github.com/ros2/common_interfaces/blob/humble/std_msgs/msg/Header.msg

```
# Standard metadata for higher-level stamped data types.
# This is generally used to communicate timestamped data
# in a particular coordinate frame.

# Two-integer timestamp that is expressed as seconds and nanoseconds.
builtin_interfaces/Time stamp

# Transform frame with which this data is associated.
string frame_id
```

- header.msg가 없으면 “msg does not have header”라는 에러가 뜬다.
 - 그러면 어떻게 이 메시지를 넣을 수 있을까?

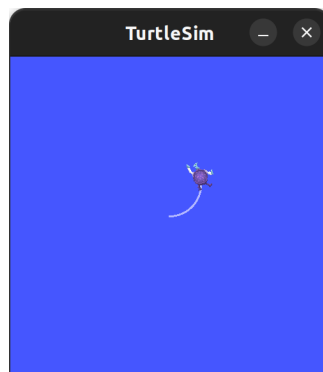
```
ros2 topic delay /(TOPIC_NAME)
```

```
jyb@jyb: ~
jyb@jyb:~$ ros2 topic delay /turtle1/cmd_vel
msg does not have header
jyb@jyb:~$
```

토픽 퍼블리시

- ros2 topic pub —once <topic_name> <msg_type> “<args>”
 - —once 옵션으로 한번의 퍼블리시만 수행한다.
 - 아래 명령어로 병진 속도 linear.x 값 2.0 m/s, 회전 속도 angular.z 값 1.8 rad/s를 입력하였다.

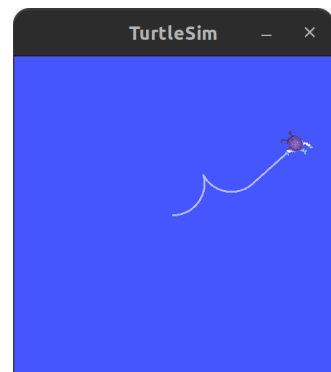
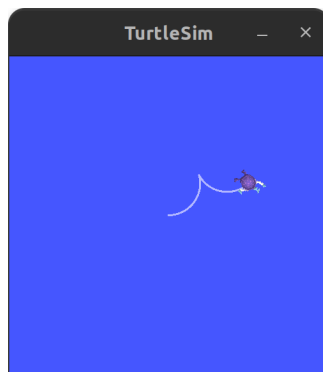
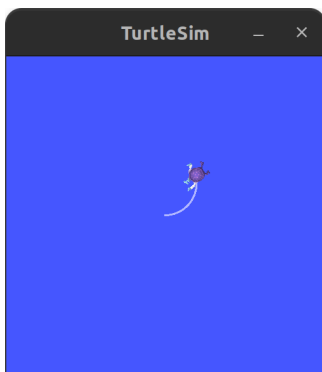
```
ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x: 2.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 1.8}}"
```



```
ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 1.8}}"
```

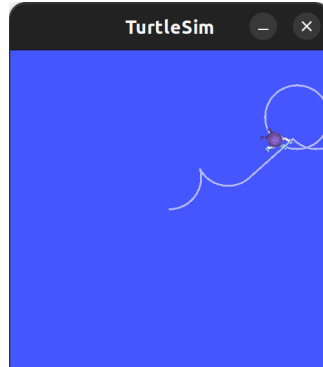
```
ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x: 0.0, y: 2.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 1.8}}"
```

```
ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x: 0.0, y: 2.0, z: 0.0}, angular: {x: 0.0, y: 2.0, z: 0.0}}"
```



- `ros2 topic pub --rate 1 <topic_name> <msg_type> "<args>"`
 - `--rate 1` 옵션으로 1 Hz 주기로 계속 퍼블리시할 수 있다.

```
ros2 topic pub --rate 1 /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x: 2.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 1.8}}
```

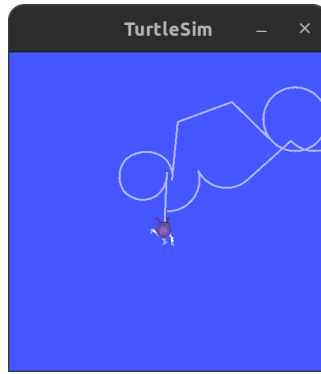


bag 기록

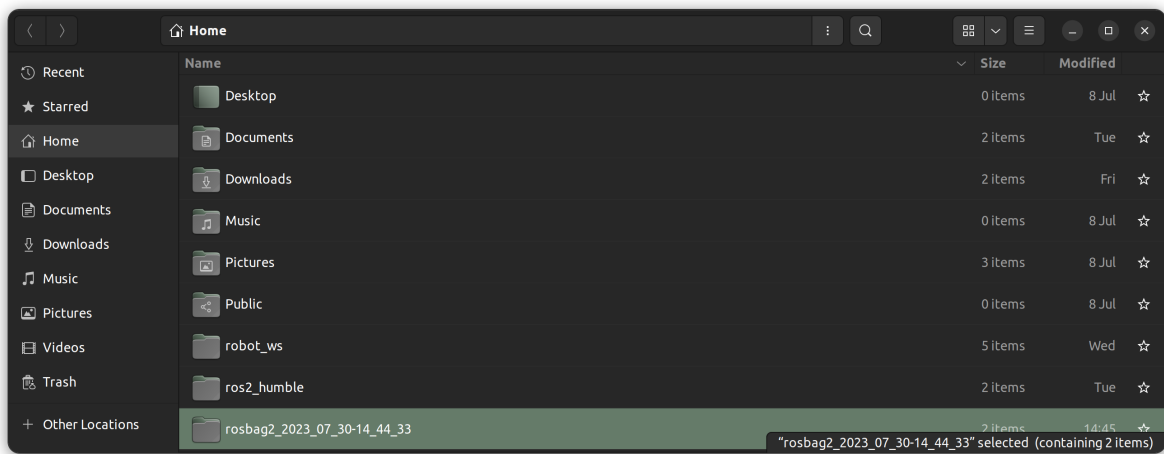
- “`ros2 bag record`” 명령으로 퍼블리시되는 토픽을 파일 형태로 저장할 수 있다. 그리고 저장된 토픽은 다시 불러와 이전과 동일한 주기로 재생할 수 있다.
 - `record` 명령을 보내고 `turtle_teleop_key`로 이동시켜보았다.

```
ros2 bag record /turtle1/cmd_vel
```

```
jyb@jyb: ~
^Cjyb@jyb:~$ ros2 bag record /turtle1/cmd_vel
[INFO] [1690695873.933714055] [rosvbag2_recorder]: Press SPACE for pausing/resumi
ng
[INFO] [1690695873.935166504] [rosvbag2_storage]: Opened database 'rosvbag2_2023_0
7_30-14_44_33/rosvbag2_2023_07_30-14_44_33_0.db3' for READ_WRITE.
[INFO] [1690695873.936989450] [rosvbag2_recorder]: Listening for topics...
[INFO] [1690695873.936998120] [rosvbag2_recorder]: Event publisher thread: Starti
ng
[WARN] [1690695873.937278071] [ROSVBAG2_TRANSPORT]: Hidden topics are not recorde
d. Enable them with --include-hidden-topics
[INFO] [1690695873.937319874] [rosvbag2_recorder]: Recording...
[INFO] [1690695874.041226264] [rosvbag2_recorder]: Subscribed to topic '/turtle1/
cmd_vel'
[INFO] [1690695874.041317966] [rosvbag2_recorder]: All requested topics are subsc
ribed. Stopping discovery...
[INFO] [1690695927.186468492] [rosvbag2_cpp]: Writing remaining messages from cac
he to the bag. It may take a while
[INFO] [1690695927.187773268] [rosvbag2_recorder]: Event publisher thread: Exitin
g
[INFO] [1690695927.187895554] [rosvbag2_recorder]: Recording stopped
[INFO] [1690695927.189201002] [rosvbag2_recorder]: Recording stopped
```

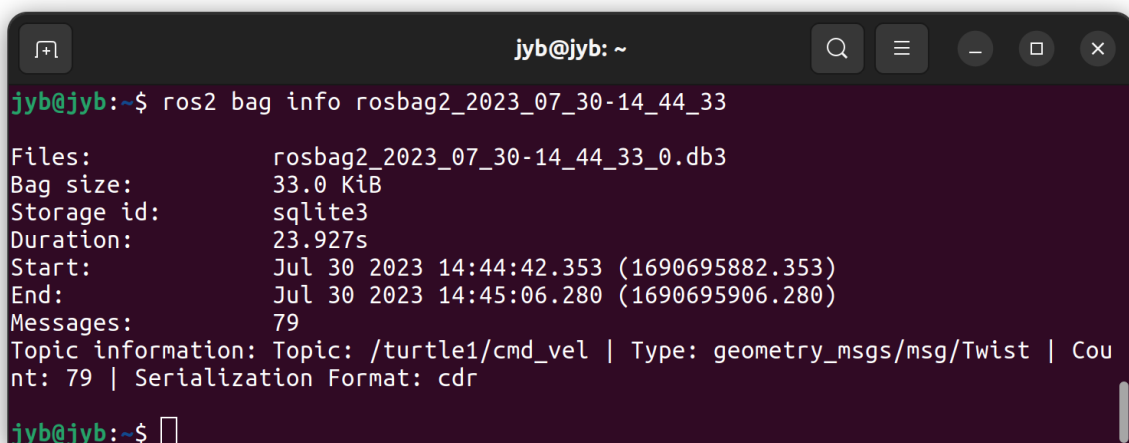


- record를 중지 (창에서 ctrl + c) 하니 "rosbag2_2023_07_30-14_44_33" 폴더가 생성되었다.



bag 정보

```
ros2 bag info rosbag2_2023_07_30-14_44_33
```



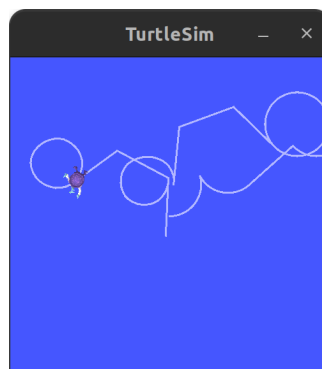
bag 재생

- record를 시작한 시점부터 토픽이 저장된다. 그 이전에 실행한 것을 저장되지 않는다.

```
ros2 bag play rosbag2_2023_07_30-14_44_33
```

```
jyb@jyb: ~
jyb@jyb:~$ ros2 bag play rosbag2_2023_07_30-14_44_33
[INFO] [1690696211.632803812] [rosbag2_storage]: Opened database 'rosbag2_2023_07_30-14_44_33/rosbag2_2023_07_30-14_44_33_0.db3' for READ_ONLY.
[INFO] [1690696211.632846214] [rosbag2_player]: Set rate to 1
[INFO] [1690696211.634686247] [rosbag2_player]: Adding keyboard callbacks.
[INFO] [1690696211.634708324] [rosbag2_player]: Press SPACE for Pause/Resume
[INFO] [1690696211.634714080] [rosbag2_player]: Press CURSOR_RIGHT for Play Next
Message
[INFO] [1690696211.634718492] [rosbag2_player]: Press CURSOR_UP for Increase Rate 10%
[INFO] [1690696211.634722445] [rosbag2_player]: Press CURSOR_DOWN for Decrease Rate 10%
[INFO] [1690696211.634961596] [rosbag2_storage]: Opened database 'rosbag2_2023_07_30-14_44_33/rosbag2_2023_07_30-14_44_33_0.db3' for READ_ONLY.
jyb@jyb:~$
```

- 기존 turtlesim node에서 play를 했을 때



- 기존 node를 끄고 새로운 node를 실행시킨 후 play 했을 때



ros2 topic help 창


```
jyb@jyb: ~  
jyb@jyb:~$ ros2 topic  
usage: ros2 topic [-h] [--include-hidden-topics]  
                Call `ros2 topic <command> -h` for more detailed usage. ...  
  
Various topic related sub-commands  
  
options:  
  -h, --help                show this help message and exit  
  --include-hidden-topics    Consider hidden topics as well  
  
Commands:  
  bw      Display bandwidth used by topic  
  delay   Display delay of topic from timestamp in header  
  echo    Output messages from a topic  
  find    Output a list of available topics of a given type  
  hz      Print the average publishing rate to screen  
  info    Print information about a topic  
  list    Output a list of available topics  
  pub     Publish a message to a topic  
  type    Print a topic's type  
  
  Call `ros2 topic <command> -h` for more detailed usage.  
jyb@jyb:~$
```