

connecting snark and ROS: points-to-ros/points-from-ros

points-to-ros and points-from-ros are utilities for publishing and receiving PointCloud2 message on ROS.

setup

To build them you need to set "snark_build_ros" to ON in snark cmake.

we use snark-graphics-test-pattern to generate some sample points in a cube:

```
snark-graphics-test-pattern cube 100000 0.1 0.01 >cube.csv
```

Here is the output: [cube.csv](#)

To run ROS, you need to setup the environment and run roscore:

```
source /opt/ros/kinetic/setup.bash
roscore
```

points-from-ros

This utility subscribes to the specified topic and receives PointCloud2 messages, then it writes the point data as csv or binary to stdout.

Either --binary or --format option must be specified, which sets the output to be binary or ascii csv respectively.

The field names and message format are embedded in the message, the format is used for conversion.

You can use --output-fields or --output-format to get the field names and message format from message (the publisher must be running).

```
source /opt/ros/kinetic/setup.bash
points-from-ros --topic "/points1" --output-fields
points-from-ros --topic "/points1" --output-format
#ascii
points-from-ros --topic "/points1" --fields x,y,z,r,g,b --format 3d,3ub | view-points --fields x,y,z,r,g,b
#binary
points-from-ros --topic "/points1" --fields x,y,z,r,g,b --binary 3d,3ub | view-points --fields x,y,z,r,g,b --
binary 3d,3ub
```

points-to-ros

This utility reads binary or ascii csv data from stdin and publishes it as PointCloud2 message on ROS.

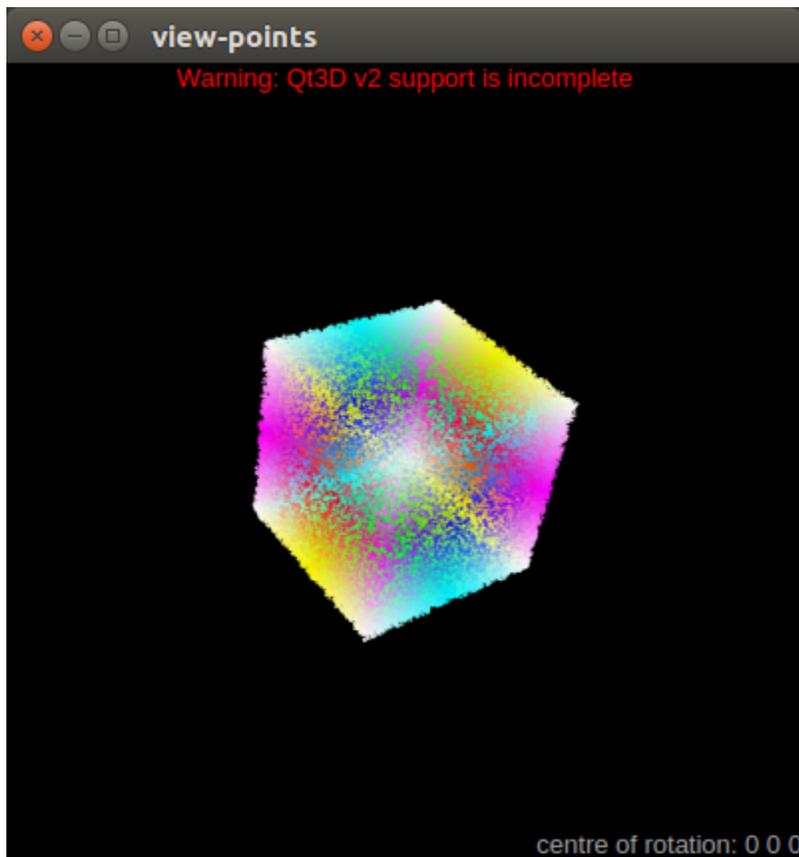
Either --binary or --format option must be specified, which indicates whether input is binary or ascii.

The --fields options specifies the field names for one point in the message.

If a field named block is present it will be used for breaking records into separate messages, records with the same block number will be grouped into one message. When no such field is present it will read the stdin until EOF and then send one message.

The --hang-on option delays the points-to-ros exit, so that the clients can receive all the data on the last message.

```
#ascii
cat cube.csv | points-to-ros --topic "/points1" --fields x,y,z,r,g,b,a --format 3d,3ub,ub --hang-on
#binary
cat cube.csv | csv-to-bin 3d,3ub,ub | points-to-ros --topic "/points1" --fields x,y,z,r,g,b,a --binary 3d,3ub,
ub --hang-on
```



ros-bag-to-bin

this utility can directly cat binary data from a ros bag file

```
ros-bag-to-bin -h
ros-bag-to-bin [-h] [--timestamp] [--block] file topic size
e.g.
ros-bag-to-bin "pointcloud.bag" "/velodyne_points" $(csv-size 4f) --timestamp --block | csv-from-bin t,ui,4f |
head
```