Please ask about problems and questions regarding this tutorial on answers.ros.org (http://answers.ros.org). Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

Recording and playing back data

Description: This tutorial will teach you how to record data from a running ROS system into a .bag file, and then to play back the data to produce similar behavior in a running system.

Keywords: data, rosbag, record, play, info, bag

Tutorial Level: BEGINNER

Next Tutorial: Getting started with roswtf (/ROS/Tutorials/Getting%20started%20with%20roswtf)

Contents

1. Recording data (creating a bag file)

This section of the tutorial will instruct you how to record topic data from a running ROS system. The topic data will be accumulated in a bag file.

First, execute the following two commands:

```
roscore
rosrun turtlesim turtlesim_node
rosrun turtlesim turtle_teleop_key
```

This will start two nodes - the turtlesim visualizer and a node that allows for the keyboard control of turtlesim using the arrows keys on the keyboard. If you select the terminal window from which you launched turtle_keyboard, you should see something like the following:

```
Reading from keyboard
---------------------------
Use arrow keys to move the turtle.
```

Pressing the arrow keys on the keyboard should cause the turtle to move around the screen. Note that to move the turtle you must have the terminal from which you launched turtlesim selected and not the turtlesim window.

1.1 Recording all published topics

First lets examine the full list of topics that are currently being published in the running system. To do this, open a new terminal and execute the command:

```
rostopic list -v
```

This should yield the following output:

```
Published topics:
 * /turtle1/color_sensor [turtlesim/Color] 1 publisher
 * /turtle1/cmd_vel [geometry_msgs/Twist] 1 publisher
 * /rosout [rosgraph_msgs/Log] 2 publishers
 * /rosout_agg [rosgraph_msgs/Log] 1 publisher
 * /turtle1/pose [turtlesim/Pose] 1 publisher

Subscribed topics:
 * /turtle1/cmd_vel [geometry_msgs/Twist] 1 subscriber
 * /rosout [rosgraph_msgs/Log] 1 subscriber
```

The list of published topics are the only message types that could potentially be recorded in the data log file, as only published messages are recorded. The topic /turtle1/cmd_vel is the command message published by teleop_turtle that is taken as input by the turtlesim process. The messages /turtle1/color_sensor and /turtle1/pose are output messages published by turtlesim.

We now will record the published data. Open a new terminal window. In this window run the following commands:

```
mkdir ~/bagfiles
cd ~/bagfiles
rosbag record -a
```

Here we are just making a temporary directory to record data and then running **rosbag record** with the option -a, indicating that all published topics should be accumulated in a bag file.

Move back to the terminal window with turtle_teleop and move the turtle around for 10 or so seconds.

In the window running rosbag record exit with a Ctrl-C. Now examine the contents of the directory ~/bagfiles. You should see a file with a name that begins with the year, data, and time and the suffix .bag. This is the bag file that contains all topics published by any node in the time that **rosbag record** was running.

2. Examining and playing the bag file

Now that we've recorded a bag file using **rosbag record** we can examine it and play it back using the commands **rosbag info** and **rosbag play**. First we are going to see what's recorded in the bag file. We can do the **info** command -- this command checks the contents of the bag file without playing it back. Execute the following command from the bagfiles directory:

```
rosbag info <your bagfile>
```

You should see something like:

```
path:        2014-12-10-20-08-34.bag
version:     2.0
duration:    1:38s (98s)
start:       Dec 10 2014 20:08:35.83 (1418270915.83)
end:         Dec 10 2014 20:10:14.38 (1418271014.38)
size:        865.0 KB
messages:    12471
compression: none [1/1 chunks]
types:       geometry_msgs/Twist [9f195f881246fdfa2798d1d3eebca84a]
             rosgraph_msgs/Log   [acffd30cd6b6de30f120938c17c593fb]
             turtlesim/Color     [353891e354491c51aabe32df673fb446]
             turtlesim/Pose      [863b248d5016ca62ea2e895ae5265cf9]
topics:      /rosout                   4 msgs    : rosgraph_msgs/Log   (2 connections)
             /turtle1/cmd_vel        169 msgs    : geometry_msgs/Twist
             /turtle1/color_sensor  6149 msgs    : turtlesim/Color
             /turtle1/pose          6149 msgs    : turtlesim/Pose
```

This tells us topic names and types as well as the number (count) of each message topic contained in the bag file. We can see that of the topics being advertised that we saw in the **rostopic** output, four of the five were actually published over our recording interval. As we ran **rosbag record** with the -a flag it recorded all messages published by all nodes.

The next step in this tutorial is to replay the bag file to reproduce behavior in the running system. First kill the teleop program that may be still running from the previous section - a Ctrl-C in the terminal where you started turtle_teleop_key. Leave turtlesim running. In a terminal window run the following command in the directory where you took the original bag file:

```
rosbag play <your bagfile>
```

In this window you should immediately see something like:

```
[ INFO] [1418271315.162885976]: Opening 2014-12-10-20-08-34.bag

Waiting 0.2 seconds after advertising topics... done.

Hit space to toggle paused, or 's' to step.
```

In its default mode rosbag play will wait for a certain period (.2 seconds) after advertising each message before it actually begins publishing the contents of the bag file. Waiting for some duration allows any subscriber of a message to be alerted that the message has been advertised and that messages may follow. If **rosbag play** publishes messages immediately upon advertising, subscribers may not receive the first several published messages. The waiting period can be specified with the -d option.

Eventually the topic /turtle1/cmd_vel will be published and the turtle should start moving in turtlesim in a pattern similar to the one you executed from the teleop program. The duration between running **rosbag play** and the turtle moving should be approximately equal to the time between the original **rosbag record** execution and issuing the commands from the keyboard in the beginning part of the tutorial. You can have **rosbag play** not start at the beginning of the bag file but instead start some duration past the beginning using the -s argument. A final option that may be of interest is the -r option, which allows you to change the rate of publishing by a specified factor. If you execute:

```
rosbag play -r 2 <your bagfile>
```

You should see the turtle execute a slightly different trajectory - this is the trajectory that would have resulted had you issued your keyboard commands twice as fast.

3. Recording a subset of the data

When running a complicated system, such as the pr2 software suite, there may be hundreds of topics being published, with some topics, like camera image streams, potentially publishing huge amounts of data. In such a system it is often impractical to write log files consisting of all topics to disk in a single bag file. The **rosbag record** command supports logging only particular topics to a bag file, allowing a user to only record the topics of interest to them.

If any turtlesim nodes are running exit them and relaunch the keyboard teleop launch file:

```
rosrun turtlesim turtlesim_node
rosrun turtlesim turtle_teleop_key
```

In your bagfiles directory, run the following command:

```
rosbag record -O subset /turtle1/cmd_vel /turtle1/pose
```

The -O argument tells **rosbag record** to log to a file named subset.bag, and the topic arguments cause **rosbag record** to only subscribe to these two topics. Move the turtle around for several seconds using the keyboard arrow commands, and then Ctrl-C the **rosbag record**.

Now check the contents of the bag file (**rosbag info subset.bag**). You should see something like this, with only the indicated topics:

```
path:        subset.bag
version:     2.0
duration:    12.6s
start:       Dec 10 2014 20:20:49.45 (1418271649.45)
end:         Dec 10 2014 20:21:02.07 (1418271662.07)
size:        68.3 KB
messages:    813
compression: none [1/1 chunks]
types:       geometry_msgs/Twist [9f195f881246fdfa2798d1d3eebca84a]
             turtlesim/Pose      [863b248d5016ca62ea2e895ae5265cf9]
topics:      /turtle1/cmd_vel    23 msgs    : geometry_msgs/Twist
             /turtle1/pose      790 msgs    : turtlesim/Pose
```

4. The limitations of rosbag record/play

In the previous section you may have noted that the turtle's path may not have exactly mapped to the original keyboard input - the rough shape should have been the same, but the turtle may not have exactly tracked the same path. The reason for this is that the path tracked by turtlesim is very sensitive to small changes in timing in the system, and rosbag is limited in its ability to exactly duplicate the behavior of a running system in terms of when messages are recorded and processed by rosrecord, and when messages are produced and processed when using rosplay. For nodes like turtlesim, where minor timing changes in when command messages are processed can subtly alter behavior, the user should not expect perfectly mimicked behavior.

Now that you've learned how to record and play back data, let's learn how to troubleshoot with roswtf (/ROS/Tutorials/Getting%20started%20with%20roswtf).

Wiki: ROS/Tutorials/Recording and playing back data (last edited 2015-05-28 01:04:03 by MuhammedHussain (/MuhammedHussain))

(http://www.osrfoundation.org)