Using rqt\_console and roslaunch

**Description:** This tutorial introduces ROS using [rqt\_console](http://wiki.ros.org/rqt_console) and [rqt\_logger\_level](http://wiki.ros.org/rqt_logger_level) for debugging and [roslaunch](http://wiki.ros.org/roslaunch) for starting many nodes at once. If you use ROS fuerte or ealier distros where [rqt](http://wiki.ros.org/rqt) isn't fully available, please see this page with [this page](http://wiki.ros.org/ROS/Tutorials/UsingRxconsoleRoslaunch) that uses old rx based tools.  
  
**Tutorial Level:** BEGINNER  
  
**Next Tutorial:**  [Using rosed](http://wiki.ros.org/ROS/Tutorials/UsingRosEd) 

**Contents**

1. [Prerequisites rqt and turtlesim package](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#Prerequisites_rqt_and_turtlesim_package)
2. [Using rqt\_console and rqt\_logger\_level](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#Using_rqt_console_and_rqt_logger_level)
   1. [Quick Note about logger levels](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#Quick_Note_about_logger_levels)
   2. [Using roslaunch](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#Using_roslaunch)
   3. [The Launch File](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#The_Launch_File)
   4. [The Launch File Explained](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#The_Launch_File_Explained)
   5. [roslaunching](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#roslaunching)

Prerequisites rqt and turtlesim package

The tutorial uses both the rqt and turtlesim packages. To do this tutorial, please install both packages, if you have not yet done so.

$ sudo apt-get install ros-<distro>-rqt ros-<distro>-rqt-common-plugins ros-<distro>-turtlesim

Replace <distro> with the name of your [ROS distribution](http://wiki.ros.org/Distributions) (e.g. indigo, jade, kinetic, lunar...).

**NOTE:** you may have already built rqt and turtlesim for one of the previous tutorials. If you are not sure, installing them again will not hurt anything.

Using rqt\_console and rqt\_logger\_level

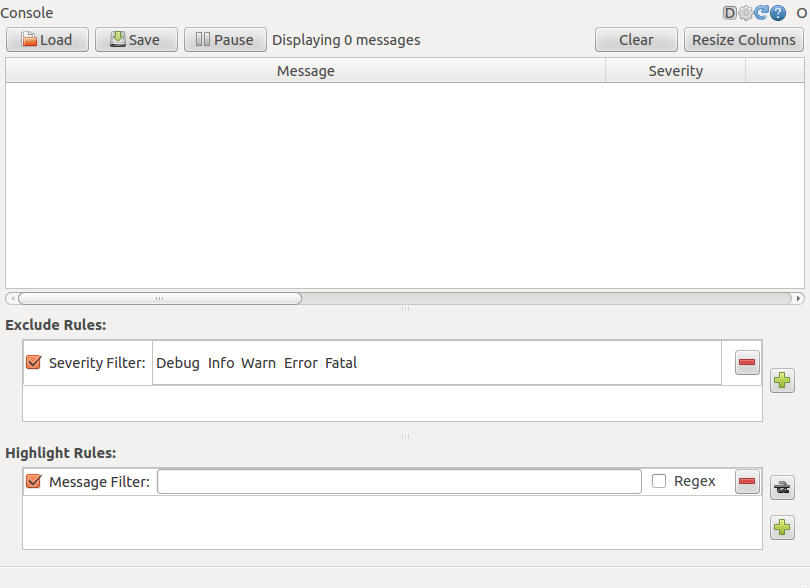
rqt\_console attaches to ROS's logging framework to display output from nodes. rqt\_logger\_level allows us to change the verbosity level (DEBUG, WARN, INFO, and ERROR) of nodes as they run.

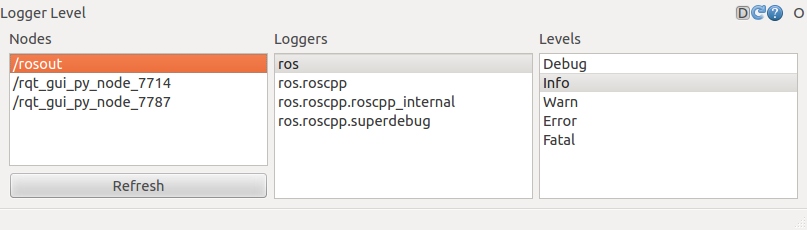
Now let's look at the turtlesim output in rqt\_console and switch logger levels in rqt\_logger\_level as we use turtlesim. Before we start the turtlesim, **in two new terminals** start rqt\_console and rqt\_logger\_level:

$ rosrun rqt\_console rqt\_console

$ rosrun rqt\_logger\_level rqt\_logger\_level

You will see two windows popup:

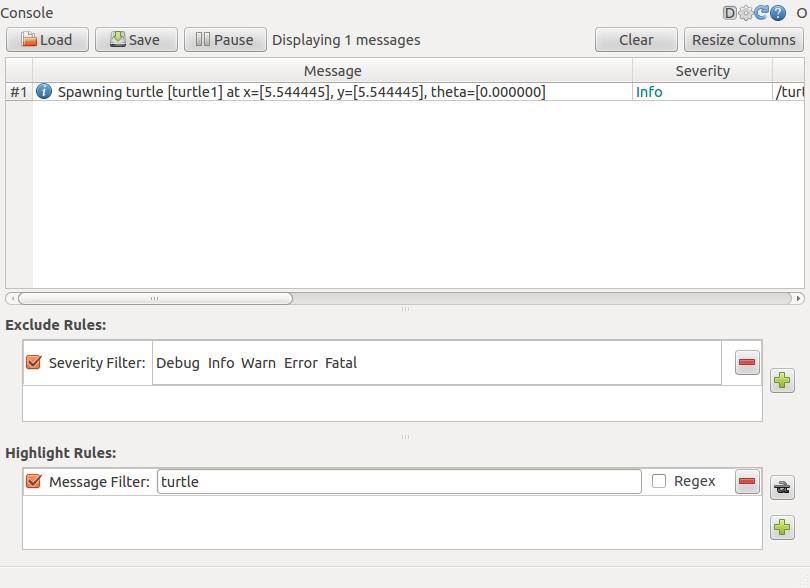




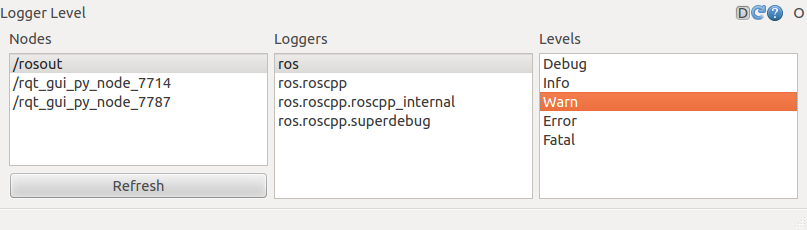
Now let's start turtlesim in a **new terminal**:

$ rosrun turtlesim turtlesim\_node

Since the default logger level is INFO you will see any info that the turtlesim publishes when it starts up, which should look like:



Now let's change the logger level to Warn by refreshing the nodes in the rqt\_logger\_level window and selecting Warn as shown below:



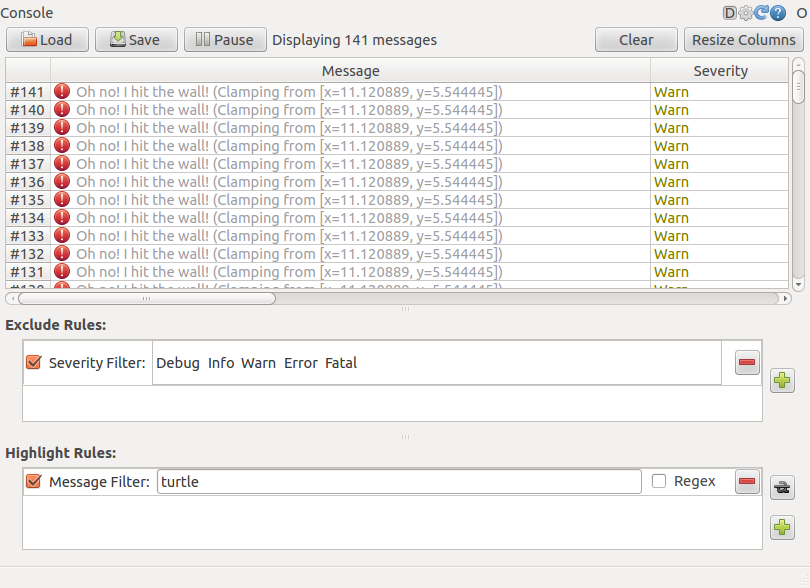
Now let's run our turtle into the wall and see what is displayed in our rqt\_console:

*For ROS Hydro and later,*

* rostopic pub /turtle1/cmd\_vel geometry\_msgs/Twist -r 1 -- '{linear: {x: 2.0, y: 0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 0.0}}'

*For ROS Groovy and earlier,*

* rostopic pub /turtle1/command\_velocity turtlesim/Velocity -r 1 -- 2.0 0.0



Quick Note about logger levels

Logging levels are prioritized in the following order:

Fatal

Error

Warn

Info

Debug

Fatal has the highest priority and Debug has the lowest. By setting the logger level, you will get all messages of that priority level or higher. For example, by setting the level to Warn, you will get all Warn, Error, and Fatal logging messages.

Let's Ctrl-C our turtlesim and let's use roslaunch to bring up multiple turtlesim nodes and a mimicking node to cause one turtlesim to mimic another:

Using roslaunch

roslaunch starts nodes as defined in a launch file.

Usage:

$ roslaunch [package] [filename.launch]

First go to the beginner\_tutorials package we [created](http://wiki.ros.org/ROS/Tutorials/CreatingPackage) and [built](http://wiki.ros.org/ROS/Tutorials/BuildingPackages) earlier:

$ roscd beginner\_tutorials

If roscd says something similar to *roscd: No such package/stack 'beginner\_tutorials'*, you will need to source the environment setup file like you did at the end of the [create\_a\_workspace](http://wiki.ros.org/catkin/Tutorials/create_a_workspace) tutorial:

$ cd ~/catkin\_ws

$ source devel/setup.bash

$ roscd beginner\_tutorials

Then let's make a launch directory:

$ mkdir launch

$ cd launch

* NOTE: The directory to store launch files doesn't necessarily have to be named launch. In fact you don't even need to store them in a directory. roslaunch command automatically looks into the passed package and detects available launch files. However, this is considered good practice.

The Launch File

Now let's create a launch file called turtlemimic.launch and paste the following:

[Toggle line numbers](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch)

[1](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_1) <launch>

[2](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_2)

[3](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_3) <group ns="turtlesim1">

[4](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_4) <node pkg="turtlesim" name="sim" type="turtlesim\_node"/>

[5](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_5) </group>

[6](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_6)

[7](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_7) <group ns="turtlesim2">

[8](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_8) <node pkg="turtlesim" name="sim" type="turtlesim\_node"/>

[9](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_9) </group>

[10](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_10)

[11](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_11) <node pkg="turtlesim" name="mimic" type="mimic">

[12](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_12) <remap from="input" to="turtlesim1/turtle1"/>

[13](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_13) <remap from="output" to="turtlesim2/turtle1"/>

[14](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_14) </node>

[15](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_15)

[16](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-91a3946a9c4cf7301bb55ec0c3f8a77f6c8f9777_16) </launch>

The Launch File Explained

Now, let's break the launch xml down.

[Toggle line numbers](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch)

[1](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-21ef414cf4c910bb1286ff2aedfe349a32a099b9_1) <launch>

Here we start the launch file with the launch tag, so that the file is identified as a launch file.

[Toggle line numbers](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch)

[3](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-0975bc12bed743bd6d6b7cf5af4a7bc4bf2fdd64_3) <group ns="turtlesim1">

[4](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-0975bc12bed743bd6d6b7cf5af4a7bc4bf2fdd64_4) <node pkg="turtlesim" name="sim" type="turtlesim\_node"/>

[5](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-0975bc12bed743bd6d6b7cf5af4a7bc4bf2fdd64_5) </group>

[6](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-0975bc12bed743bd6d6b7cf5af4a7bc4bf2fdd64_6)

[7](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-0975bc12bed743bd6d6b7cf5af4a7bc4bf2fdd64_7) <group ns="turtlesim2">

[8](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-0975bc12bed743bd6d6b7cf5af4a7bc4bf2fdd64_8) <node pkg="turtlesim" name="sim" type="turtlesim\_node"/>

[9](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-0975bc12bed743bd6d6b7cf5af4a7bc4bf2fdd64_9) </group>

Here we start two groups with a namespace tag of turtlesim1 and turtlesim2 with a turtlesim node with a name of sim. This allows us to start two simulators without having name conflicts.

[Toggle line numbers](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch)

[11](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-78308b822a594630211cae0b2b508b405d07b108_11) <node pkg="turtlesim" name="mimic" type="mimic">

[12](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-78308b822a594630211cae0b2b508b405d07b108_12) <remap from="input" to="turtlesim1/turtle1"/>

[13](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-78308b822a594630211cae0b2b508b405d07b108_13) <remap from="output" to="turtlesim2/turtle1"/>

[14](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-78308b822a594630211cae0b2b508b405d07b108_14) </node>

Here we start the mimic node with the topics input and output renamed to turtlesim1 and turtlesim2. This renaming will cause turtlesim2 to mimic turtlesim1.

[Toggle line numbers](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch)

[16](http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch#CA-e33435967e0cc6e3273fdf1ce957aba2daaf5023_16) </launch>

This closes the xml tag for the launch file.

roslaunching

Now let's roslaunch the launch file:

$ roslaunch beginner\_tutorials turtlemimic.launch

Two turtlesims will start and in a **new terminal** send the rostopic command:

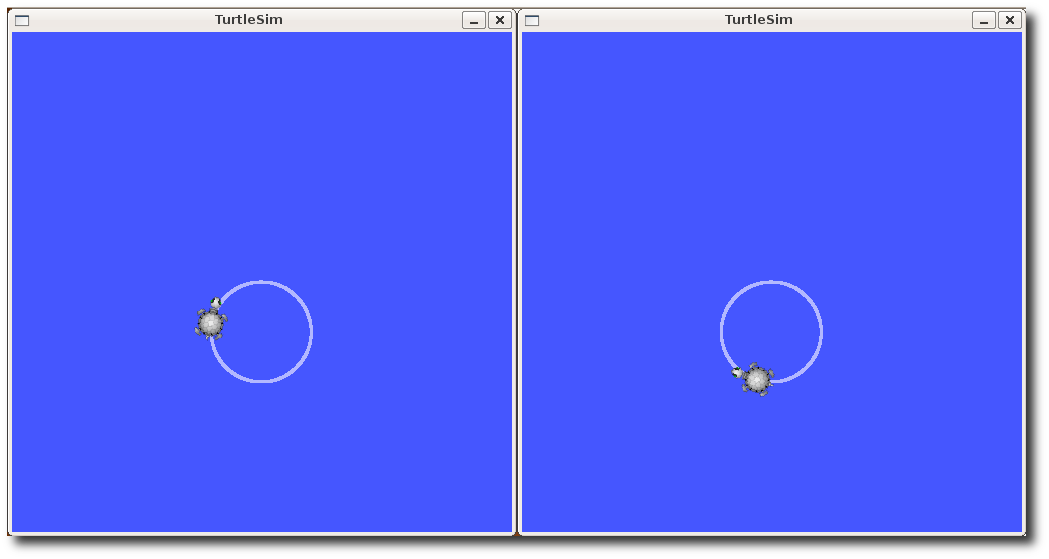
*For ROS Hydro and later,*

* $ rostopic pub /turtlesim1/turtle1/cmd\_vel geometry\_msgs/Twist -r 1 -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, -1.8]'

*For ROS Groovy and earlier,*

* $ rostopic pub /turtlesim1/turtle1/command\_velocity turtlesim/Velocity -r 1 -- 2.0 -1.8

You will see the two turtlesims start moving even though the publish command is only being sent to turtlesim1.

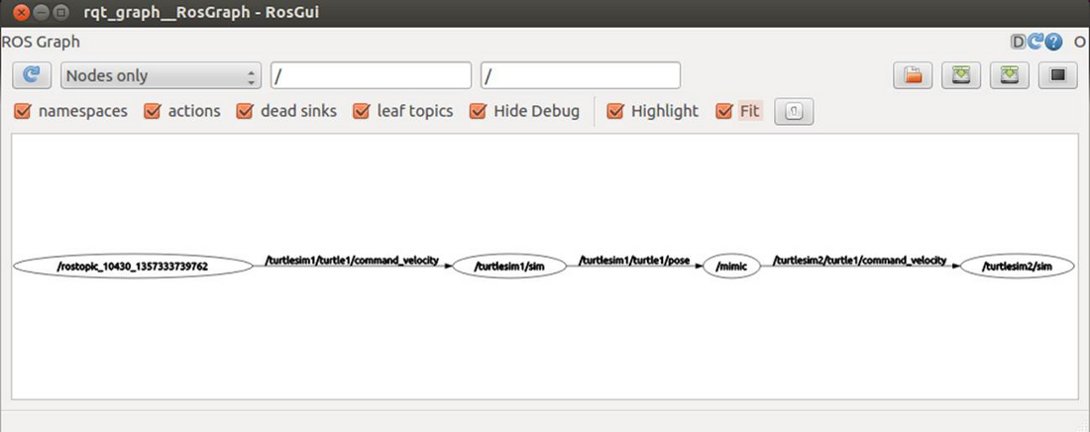


We can also use [rqt\_graph](http://wiki.ros.org/rqt_graph) to better understand what our launch file did. Run [rqt](http://wiki.ros.org/rqt)'s main window and select *Plugins > Introspection > Node Graph*:

$ rqt

Or simply:

$ rqt\_graph



Now that you have successfully used rqt\_console and roslaunch, let's learn about [editor options for ROS](http://wiki.ros.org/ROS/Tutorials/UsingRosEd). You can Ctrl-C