ROS Indigo Cheatsheet

Filesystem Management Tools

**rosrun**
Run a ROS package's executable with minimal typing.

Usage:
```
rosrun [package name executable.name]
```

Example (runs turtlebot3):
```
rosrun turtlebot3 turtlebot_node
```

**roslaunch**
Start a roscore (if needed), local nodes, remote nodes via SSH, and sets parameter server connections.

Examples:
```
Launch a file to a package:
$ roslaunch package name file name.launch

Launch on a different port:
$ roslaunch -p 1234 package name file name.launch

Launch on the local nodes:
$ roslaunch local package name file name.launch
```

Logging Tools

**rosbag**
A tool for recording and playing back of ROS topics.

Commands:
- record
- replay
- compress
- decompress
- filter

Examples:
- Record select topics:
  ```
  $ rosbag record topic1 topic2
  $ rosbag replay -a demo_log.bag
  $ rosbag replay -a demo_log.bag
  $ rosbag replay -a demo_log.bag
  ```

Integration and Command Tools

**rosmsg**
Displays Message/Service (msg/srv) data structure definitions.

Commands:
- show
- list
- info
- packages

Examples:
- Display the Pose msg:
  ```
  $ rosmsg show Pose
  ```
- List the messages in the sensor_msgs package:
  ```
  $ rosmsg list sensor_msgs
  ```

**rosnode**
Displays debugging information about ROS nodes, including publications, subscriptions and connections.

Examples:
- Test connectivity to node:
  ```
  $ rosnode ping
  ```
- Print information about a node:
  ```
  $ rosnode info
  ```

**rosnodekill**
Kill a running node.

Examples:
- Kill all nodes:
  ```
  $ rosnodekill
  ```
- Kill node on a machine:
  ```
  $ rosnodekill machineName
  ```

**rosrun**
A tool for launching and querying ROS services.

Commands:
- list
- node
- call
- args
- type
- uri
- find

Examples:
- Call a service from the command-line:
  ```
  $ rosrun your_package your_service_name 1 2
  ```
- Display all services of a particular type:
  ```
  $ rosrun your_package your_service_name
  ```

**rostopic**
A tool for displaying information about ROS topics, including publishers, subscribers, publishing rate, and messages.

Commands:
- list
- echo
display publish rate of topic
- info
- list all published topics
- pub
- type
- type
- list
- echo
- info
- list all published topics
- pub
- type
- type
- list
- echo
- info
- list all published topics
- pub
- type
- type
ROS Indigo Cheatsheet

Logging Tools

rqt_console
A tool for displaying and filtering messages published on a topic.
Usage:
$ rqt.console

rqt_bag
A tool for reading, inspecting, and replaying bag files.
Usage, viewing:
$ rqt_bag bag_file.bag
Usage, bagging:
$ rqt_bag --record the_big_red_record_bag.bag

rqt_logger_level
Change the logger level of ROS nodes. This will increase or decrease the information they log to the screen and replays.
Usage:
viewing $ rqt_logger_level

Introversion & Command Tools

rqt_topic
A tool for viewing published topics in real time.
Usage:
$ rqt

rqt_mng, rqt_srv, and rqt_action
A tool for visualizing available maps, services, and actions.
Usage:
$ rqt

rqt_publisher, and rqt_service.caller
Tools for publishing messages and calling services.
Usage:
$ rqt
Plugin Menu->Topic->Message Publisher
Plugin Menu->Service->Service Caller

rqt_graph, and rqtdep
Tools for displaying graphs of running ROS nodes with connecting topics and package dependencies respectively.
Usage:
$ rqt_graph
$ rqtdep

rqt_top
A tool for specific process monitoring.
Usage:
$ rqt
Plugin Menu->Introspection->Process Monitor

rqt_reconfigure
A tool for dynamically reconfiguring ROS parameters.
Usage:
$ rqt
Plugin Menu->Configuration->Dynamic Reconfigure

Development Environments

rqt.shell, and rqt.py.console
Two tools for accessing an xterm shell and python console respectively.
Usage:
$ rqt
Plugin Menu->Miscellaneous Tools->Shell
Plugin Menu->Miscellaneous Tools->Python Console

Data Visualization Tools

tf_echo
A tool that prints the information about a particular transformation between a source.frame and a target.frame.
Usage:
$ rosrun tf tf_echo <source.frame> <target.frame>

Examples:
To echo the transform between /map and /odom:
$ rosrun tf tf_echo /map /odom

view_frames
A tool for visualizing the full tree of coordinate transforms.
Usage:
$ rosrun tf2tools view_frames.py
$ view_frames.pdf

rqt_plot
A tool for plotting data from ROS topic fields.
Usage:

Examples:
To graph the data in different plots:
$ rqt_plot /topic/field1 /topic/field2
To graph the data all on the same plot:
$ rqt_plot /topic/field1 /topic/field2
To graph multiple fields of a message:
$ rqt_plot /topic/field1/field2/field3

rqt_image_view
A tool for displaying image topics.
Usage:
$ rqt_image_view
ROS Indigo Catkin Workspaces

Create a catkin workspace
Setup and use a new catkin workspace from scratch.

Examples:
$ source /opt/ros/indigo/setup.bash
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/src
$ catkin_init_workspace

Checkout an existing ROS package
Get a local copy of the code for an existing package and keep it up to date using wstool.

Examples:
$ cd ~/catkin_ws/src
$ wstool init
$ wstool set tutorials --git https://github.com/ros/ros-tutorials.git
$ wstool update

Create a new catkin ROS package
Create a new ROS catkin package in an existing workspace with catkin create package. After using this you will need to edit the CMakeLists.txt to detail how you want your package built and add information to your package.xml.

Usage:
catkin_create_pkg <package_name> [depend1] [depend2]

Examples:
$ cd ~/catkin_ws/src
$ catkin_create_pkg tutorials std_msgs rospkg rospy

Build all packages in a workspace
Use catkin_make to build all the packages in the workspace and then source the setup.sh to add the workspace to the ROS_PACKAGE_PATH.

Examples:
$ cd ~/catkin_ws
$ catkin_make
$ source devel/setup.bash

Copyright © 2015 Open Source Robotics Foundation
Copyright © 2010 Willow Garage

From:
https://robosub.eecs.wsu.edu/wiki/ - Palouse RoboSub Technical Documentation

Permanent link:
https://robosub.eecs.wsu.edu/wiki/cs/ros/cheatsheet/start

Last update: 2016/11/16 12:34