**ROS Indigo Cheatsheet**

**Filesystem Management Tools**

- `rospack`: A tool for inspecting packages.
- `rospack profile`: Prints package and plugin problems.
- `roscd`: Change directory to a package.
- `roscd/roscd`: Prints the root of the ROS.
- `rostopic`: Starts a rosnode if needed, local nodes, remote nodes via SSH, and sets parameter server connections.

**Logging Tools**

### rosbag

A tool for recording and playing back ROS topics.

- **Command**: `rosbag record`
- **Options**: `rosbag record topic1 topic2`

- **Example**: Record topics `rosbag record topic1 topic2`

### rostopic

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

- **Commands**:
  - `rostopic info`: Print information about a topic.
  - `rostopic list`: List all published topics.
  - `rostopic pub`: Publish data to topic.
  - `rostopic type`: Print topic type.

- **Example**: Publish a file to a topic:
  ```
  $ rostopic pub -r 10 / thựn_std_msgs/String hello
  ```

**Introspection and Command Tools**

### roscore

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

- **Commands**:
  - `roscore show`: Display the fields in the msg/srv.
  - `roscore list`: Display list of topics.

- **Example**: Display the fields in the msg/srv:
  ```
  $ roscore show
  ```

### rostopic

A tool for getting and setting ROS parameters on the parameter server using YAML-encoded files.

- **Commands**:
  - `rostopic set`: Set a parameter.
  - `rostopic get`: Get a parameter.

- **Example**: Get a parameter from a file:
  ```
  $ rostopic get /Motor /forward
  ```

**Start-up and Process Launch Tools**

### rosrun

Runs a ROS package's executable with minimal typing.

- **Example**:
  ```
  $ rosrun turtlesim turtlesim_node
  ```

### roslaunch

Launches a ROS node, remote nodes via SSH, and sets parameter server connections.

- **Example**:
  ```
  $ roslaunch package名 file名.launch
  ```

**Rosnode**

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

- **Commands**:
  - `rosnode info`: Print information about a node.
  - `rosnode kill`: Kill a running node.

- **Example**:
  ```
  $ rosnode info
  ```

**Roservice**

A tool for listing and querying ROS services.

- **Commands**:
  - `roservice list`: Print a list of all ROS services.
  - `roservice info`: Print information about active services.

- **Example**:
  ```
  $ roservice list
  ```

**Roservice**

A tool for setting and getting ROS parameters on the

- **Command**: `roservice set`
ROS Indigo Cheatsheet

Logging Tools

rqt_console
A tool to display and filtering messages published on topics.
Usage:
$ rqt_console

rqt_bag
A tool for streaming, inspecting, and replaying log files.
Usage, viewing:
$ rqt_bag bagfile.bag
Usage, bagging:
$ rqt_bag *press the big red record button.*

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

Introspection & Command Tools

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

rqt_msg, rqt_srv, and rqt_action
A tool for viewing available msg, srv, 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 rqt_dep
Tools for displaying graphs of running ROS nodes with connected topics and package dependencies respectively.
Usage:
$ rqt_graph
$ rqt_dep

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 tf2_views view_frames.py
$ view_frames.pdf

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

Examples:
To graph the data in different plots:
$ rqt_plot /topic1/field1 /topic2/field2
To graph the data all on the same plot:
$ rqt_plot /topic1/field1 /topic2/field2
To graph multiple fields of a message:
$ rqt_plot /topic1/field1:field2

rqt_image_view
A tool for displaying image topics.
Usage:
$ rqt_image_view

Development Environments

rqt.shell, and rqt.py_console
Two tools for accessing an stern shell and python console respectively.
Usage:
$ rqt
Plugin Menu—>Miscellaneous Tools—>Shell
Plugin Menu—>Miscellaneous Tools—>Python Console
ROS Indigo Catkin Workspaces

Create a catkin workspace

Setup and use a new catkin workspace from scratch.

Examples:

```bash
$ 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:

```bash
$ cd /catkin_ws/src
$ wstool init
$ wstool fetch tutorials --git https://github.com/ros/ros_tutorials.git
```

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:

```bash
$ catkin createPKG <package_name> [depend] [depend2]
```

Example:

```bash
$ cd /catkin_ws/src
$ catkin createPKG tutorials std_msgs rospy rosdep
```

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:

```bash
$ cd /catkin_ws
$ catkin_make
$ source devel/setup.bash
```

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