ROS Indigo Cheatsheet

Filesystem Management Tools

- **rospack**
  - A tool for inspecting package.
  - FIX - pack and pluggable problems.

- **roscd**
  - Change to a package.
  - Patch equivalent for ROS.

- **rosdep**
  - List package or stack information.

- **rosrun**
  - Run a node.

- **rosmake**
  - Build a ROS cotinuous workspace.

- **rosdep**
  - Display package structure and dependencies.

Usage:
- $ rospack find [package]
- $ roscd [package/subdir]
- $ rosdep [package/subdir]
- $ rosrun [package/subdir]
- $ rosmake [package/subdir]
- $ rospack [package] [file] [destination]
- $ rosrun install [package]
- $ rosrun start [file]
- $ rosmake [package] [depend] [...]
- $ rosdump [topic]
- $ rosdump [topic]
- $ rostopic list

- **roslaunch**
  - Start a node.
  - Example (run tutorial):
    - $ roslaunch tutorial turtlesim_node

- **rosnode**
  - Start a node if needed.
  - Example:
    - $ rosnode name executable name

- **rostopic**
  - Publish a message.
  - Example:
    - $ rostopic pub /topic_name std_msgs/String hello

- **rosrun**
  - Start a node.
  - Example:
    - $ rosrun robosub.scripts.move_straight 1 1 0.01

Logging Tools

- **rosbag**
  - A tool for recording and playing back of ROS topics.
  - Commands:
    - rosbag record
    - rosbag replay
    - rosbag compress
    - rosbag decompress
    - rosbag filter

Examples:
- rosbag record topic1 topic2
- rosbag replay -a demo.bag
- rosbag compress demo.bag
- rosbag decompress demo.bag
- rosbag filter the contents of the bag

Introspection and Command Tools

- **rosmake**
  - Displays Message/Service (msg/srv) data structure definitions.
  - Commands:
    - rosmake show
    - rosmake list
    - rosmake get
    - rosmake dump
    - rosmake delete

Examples:
- $ rosmake show
- $ rosmake list

- **rostopic**
  - Publish a message.
  - Example:
    - $ rostopic pub /topic_name std_msgs/String hello

- **rospy**
  - Publish data to a topic.
  - Example:
    - $ rospy pub /topic_name std_msgs/String hello

- **rosparam**
  - A tool for getting and setting ROS parameters.
  - Commands:
    - rosparam get
    - rosparam set
    - rosparam load
    - rosparam delete
    - rosparam list

Examples:
- $ rosparam set /topic_name string value

- **roswatch**
  - A tool for watching and copying ROS services.
  - Commands:
    - roswatch
    - roswatch test
    - roswatch service
    - roswatch node

Examples:
- $ roswatch service
- $ roswatch test

- **roscore**
  - A tool for starting and stopping ROS.
  - Commands:
    - roscore
    - roscore start
    - roscore node

Examples:
- $ roscore
- $ roscore start

- **rospy**
  - Publish data to a topic.
  - Example:
    - $ rospy pub /topic_name std_msgs/String hello

- **roslint**
  - Publish data to a topic.
  - Example:
    - $ roslint test

- **rosrun**
  - Publish data to a topic.
  - Example:
    - $ rosrung test

- **rosnode**
  - Publish data to a topic.
  - Example:
    - $ rosnode test

- **roslint**
  - Publish data to a topic.
  - Example:
    - $ roslint test

- **rosrun**
  - Publish data to a topic.
  - Example:
    - $ rosrung test

- **roslint**
  - Publish data to a topic.
  - Example:
    - $ roslint test
**ROS Indigo Cheatsheet**

**Logging Tools**
- **rqt_console**
  A tool to display and filtering messages published on topics.
  Usage: `$ rqt.console`

- **rqt_bag**
  A tool to stimulating, inspecting, and replaying log files.
  Usage, viewing: `$ rqt_bag bag-file.blogspot`
  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: viewing `$ rqt_logger_level`

**Introspetion & 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 msgs, srvs, 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 connecting topics and package dependencies respectively.
  Usage:
  - `$ rqt_graph`
  - `$ rqt_dep`

**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 tf2_tools view_frames.py`
  - `$ rviz_frames.pdf`

- **rqt_plot**
  A tool for plotting data from ROS topic fields.
  Usage:
  - To graph the data in different plots:
    - `$ rqt_plot /topic1/field1 /topic2/field2`
  - To graph the data in the same plot:
    - `$ rqt_plot /topic1/field1 /topic2/field2`
  - To graph multiple fields of a message:
    - `$ rqt_plot /topic1/field1:field2:field3`

- **rqt_image_view**
  A tool to display image topics.
  Usage:
  - `$ rqt_image_view`
ROS Indigo Catkin Workspaces

**Create a catkin workspace**

Setup and use a new catkin workspace from scratch.

Example:

```
$ 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 git://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 tool 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 [depend] [depend2]
```

Example:

```
$ cd /catkin_ws/src
$ catkin_create_pkg tutorials std_msgs rospack roscpp
```

**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
```