ROS Indigo Cheatsheet

Filesystem Management Tools

rospkg
rosinstall
runde
rosnode
rosdep
catkin
wst
rqt

Usage:
- `rosinstall [package]`
- `rosnode [package]`:
- `rosdep [package]`:
- `catkin install`:
- `rqt dep`:

Start-up and Process Launch Tools

roscore
rostest

Usage:
- `roscore`:
- `rostest`:

Logging Tools

rosbag

A tool for recording and playing back of ROS topics.
- `rosbag record`:
- `rosbag play`
- `rosbag compress`
- `rosbag decompress`
- `rosbag filter`

Examples:
- `rosbag record topic1 topic2`
- `rosbag play -a /rosout.log`
- `rosbag compress /userdata/bag`
- `rosbag decompress /userdata/bag`
- `rosbag filter [regex]`

Introspection and Command Tools

rosmg/rosr

Displays Message/Service (msg/srv) data structure definitions.
- `rosmg show`:
- `rosmg list`:
- `rosmg publish`:
- `rosmg packages`:

Examples:
- `rosmg show /msg`:
- `rosmg list /msg`:
- `rosmg publish /msg`:
- `rosmg packages`

rosservice

Displays debugging information about ROS services, including publications, subscriptions and connections.
- `rosservice list`:
- `rosservice info`
- `rosservice kill`

Examples:
- `rosservice list`:
- `rosservice info [service]`:
- `rosservice kill [service]`

rostopic

A tool for displaying information about ROS topics, including publishers, subscribers, publishing rate, and messages.
- `rostopic ls`
- `rostopic echo`
- `rostopic info`
- `rostopic list`
- `rostopic pub`
- `rostopic type`

Examples:
- `rostopic ls`
- `rostopic echo [topic]`
- `rostopic info [topic]`
- `rostopic list`
- `rostopic pub [topic]`:
- `rostopic type [topic]`

rostopIC

Publish hello at 10 Hz:
- `rostopic pub -r 10 /topicname std_msgs/String hello`

Clear the screen after each message is published:
- `rostopic echo -n /topicname`

Display messages that match a given Python expression:
- `rostopic echo --filter "a.data == 42" /topicname`

Pipe the output of rostopic to rosservice to view the msg type:
- `rostopic type /topicname | rosservice show`

rospack

A tool for getting and setting ROS parameters on the parameter server using YAML-encoded files.
- `rospack set`:
- `rospack get`:
- `rospack load`:
- `rospack delete`

Examples:
- `rospack set [package] /myparam: 10`
- `rospack get /myparam`

rospkg
dump

Examples:
- `rospkg dump`:
- `rospkg dump:yaml /myparamspace`

rosservice

A tool for listing and querying ROS services.
- `rosservice list`:
- `rosservice info`
- `rosservice kill`

Examples:
- `rosservice list`
- `rosservice info [service]`
- `rosservice kill [service]`
### 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 stimulating, 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 logfiles.

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 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
```

**rqt_top**
A tool for ROS 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 empir 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:
```
$ rosservice tf_echo <source_frame> <target_frame>
```

Examples:
```
To echo the transform between /map and /odom:
$ rosservice tf_echo /map /odom
```

**view_frames**
A tool for visualizing the full tree of coordinate transforms.

Usage:
```
$ rosservice tf_tools view_frames.py
$ sirrus 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/data1 /topic2/data2
To graph the data all on the same plot:
$ rqt_plot /topic1/data1 /topic2/data2
To graph multiple fields of a message:
$ rqt_plot /topic3/data1/data2/data3
```

**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]
```

Example:

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

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
```