ROS Indigo Cheatsheet

Filesystem Management Tools

- **rsh**: A tool for inspecting package.
- **rmap**: A tool for finding packages.
- **rsync**: Change directory to a package.
- **rmdir**: Remove a directory on the hard disk.
- **rm**: List package or package information.
- **rmt**: Open the requested RSB file in a text editor.
- **rmcp**: Copy a file from one place to another.
- **rmbs**: Install package system dependencies.
- **rmmg**: Displays errors and warnings about a running ROS system or launch file.
- **rmd**: Creates a new ROS stack.
- **rsm**: Manages many repos in workspace.
- **rsd**: Builds a ROS control workspace.
- **src**: Displays package structure and dependencies.

Usage:
```
# rmap find [package]
# roset [package/substr]
# rcmp [package/substr] -r [rw]
# rsd
# rmt [package/substr]
# rsd [package] [file]
# rsm [package/substr] [file] [destination]
# rmmg [package] [package]
# rsd or rsf [file]
# rsd [package] [package]
# rsm [package/substr] [package/substr]
# setl [list | set | update]
# create
# rsm [options]
```

Start-up and Process Launch Tools

- **roscore**: Runs a ROS package's executable with minimal typing.

Usage:
```
# roscore
```

**rosout**: Starts a rosout (if needed), locates, remote nodes via SSH, and sets parameter server.

Examples:
```
# rosrun
```

- **rosdistro**: Starts a rosdistro (if needed), locates, remote nodes via SSH, and sets parameter server.

Examples:
```
# rosdistro
```

- **rosinstall**: Starts a rosinstall (if needed), locates, remote nodes via SSH, and sets parameter server.

Examples:
```
# rosinstall
```

- **roslaunch**: Starts a roslaunch (if needed), locates, remote nodes via SSH, and sets parameter server.

Examples:
```
# roslaunch
```

- **rsmg**: Starts a rosmg (if needed), locates, remote nodes via SSH, and sets parameter server.

Examples:
```
# rsmg
```

Logging Tools

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

Commands:
```
rosdebug record
rosdebug record
rosdebug record
rosdebug record
rosdebug record
```

Examples:
```
# rosdcheck message
```

Introspection and Command Tools

**rosbag**: Displays RMSG/Service (msg/msg) data structure definitions.

Commands:
```
rosbag show
rosbag list
rosbag info
rosbag package
rosbag publish
```

Examples:
```
# Display the Pose msg:
# rosbag show Pose
```

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

Commands:
```
rosnode publish
rosnode info
rosnode machine
rosnode kill
```

Examples:
```
# Kill all nodes:
# rosmaint machine list
# rosmaint machine list
# rosmaint machine list
```

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

Commands:
```
rostopic ls
rostopic info
rostopic list
rostopic pub
rostopic topic
```

Examples:
```
# rostopic publish hello
```

Palouse RoboSub Technical Documentation - https://robosub.eecs.wsu.edu/wiki/
ROS Indigo Cheatsheet

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

- **rqt_bag**
  A tool for stimulating, inspecting, and replaying log files.
  Usage: `rqt bag`

- **rqt_logger_level**
  Change the logger level of ROS nodes. This will increase or decrease the information they log to the screen and repository.
  Usage: `rqt_logger_level`

Introspection & 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 viewing available mng, srv, and actions.
  Usage: `rqt`

Data Visualization Tools
- **tf_echo**
  A tool that prints the information about a particular transformation between a source frame and a target frame.
  Usage: `tf echo <source_frame> <target_frame>`

  Examples:
  - To echo the transform between /map and /odom:
    `tf echo /map /odom`

- **view_frames**
  A tool for visualizing the full tree of coordinate transforms.
  Usage: `view_frames`

  Examples:
  - To graph the data in different plots:
    `plot /topic1 /field1 /topic2 /field2`
  - To graph the data all on the same plot:
    `plot /topic1 /field1 /topic2 /field2`
  - To graph multiple fields of a message:
    `plot /topic /field1 /field2 /field3`

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

Development Environments
- **rqt_shell and rqt_py.console**
  Two tools for accessing an xterm shell and python console respectively.
  Usage: `rqt`
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.

Example:
$ 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 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 rospy rosuxp

Build all packages in a workspace

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

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

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