

ROS Indigo Cheatsheet

Filesystem Management Tools

rospack	A tool for inspecting packages .
rospack profile	Fixes path and pluginlib problems.
roscd	Change directory to a package.
rospd/rosd	Pushd equivalent for ROS .
rosls	Lists package or stack information.
rosed	Open requested ROS file in a text editor.
roscp	Copy a file from one place to another.
rosdep	Installs package system dependencies.
roswtf	Displays a errors and warnings about a running ROS system or launch file.
catkin_create_pkg	Creates a new ROS stack.
wstool	Manage many repos in workspace.
catkin_make	Builds a ROS catkin workspace.
rqt_dep	Displays package structure and dependencies.

Usage:

```
$ rospack find [package]
$ roscd [package[/subdir]]
$ rospd [package[/subdir] | +N | -N]
$ rosd
$ rosls [package[/subdir]]
$ rosed [package] [file]
$ roscp [package] [file] [destination]
$ rosdep install [package]
$ roswtf or roswtf [file]
$ catkin_create_pkg [package_name] [depend1]..[dependN]
$ wstool [init | set | update]
$ catkin_make
$ rqt_dep [options]
```

Start-up and Process Launch Tools

roscore

The basis [nodes](#) and programs for ROS-based systems. A roscore must be running for ROS nodes to communicate.

Usage:

```
$ roscore
```

roslaunch

Runs a ROS package's executable with minimal typing.

Usage:

```
$ roslaunch package_name executable_name
```

Example (runs [turtlesim](#)):

```
$ roslaunch turtlesim turtlesim_node
```

roslaunch

Starts a roscore (if needed), [local nodes](#), [remote nodes](#) via SSH, and sets parameter server [parameters](#).

Examples:

```
Launch a file in 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
```

Introspection and Command Tools

roscd

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

Commands:

roscd ping	Test connectivity to node.
roscd list	List active nodes.
roscd info	Print information about a node.
roscd machine	List nodes running on a machine.
roscd kill	Kill a running node.

Examples:

```
Kill all nodes:
$ roscd kill -a
List nodes on a machine:
$ roscd machine aqy.local
Ping all nodes:
$ roscd ping --all
```

rostopic

A tool for displaying information about ROS [topics](#), including publishers, subscribers, publishing rate, and messages.

Commands:

rostopic bw	Display bandwidth used by topic.
rostopic echo	Print messages to screen.
rostopic find	Find topics by type.
rostopic hz	Display publishing rate of topic.
rostopic info	Print information about an active topic.
rostopic list	List all published topics.
rostopic pub	Publish data to topic.
rostopic type	Print topic type.

Examples:

```
Publish hello at 10 Hz:
$ rostopic pub -r 10 /topic_name std_msgs/String hello
Clear the screen after each message is published:
$ rostopic echo -c /topic_name
Display messages that match a given Python expression:
$ rostopic echo --filter "m.data=='foo'" /topic_name
Pipe the output of rostopic to rosmmsg to view the msg type:
$ rostopic type /topic_name | rosmmsg show
```

rosservice

A tool for listing and querying ROS services.

Commands:

rosservice list	Print information about active services.
rosservice node	Print name of node providing a service.
rosservice call	Call the service with the given args.
rosservice args	List the arguments of a service.
rosservice type	Print the service type.
rosservice uri	Print the service ROSRPC uri.
rosservice find	Find services by service type.

Examples:

```
Call a service from the command-line:
$ rosservice call /add_two_ints 1 2
Pipe the output of rosservice to rossrv to view the srv type:
$ rosservice type add_two_ints | rossrv show
Display all services of a particular type:
$ rosservice find rospy_tutorials/AddTwoInts
```

roscd

A tool for getting and setting ROS [parameters](#) on the parameter server using YAML-encoded files.

Commands:

roscd set	Set a parameter.
roscd get	Get a parameter.
roscd load	Load parameters from a file.
roscd dump	Dump parameters to a file.
roscd delete	Delete a parameter.
roscd list	List parameter names.

Examples:

```
List all the parameters in a namespace:
$ roscd list /namespace
Setting a list with one as a string, integer, and float:
$ roscd set /foo ["'1'", 1, 1.0]"
Dump only the parameters in a specific namespace to file:
$ roscd dump yml /namespace
```

rosmmsg/rossrv

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

Commands:

rosmmsg show	Display the fields in the msg/srv.
rosmmsg list	Display names of all msg/srv.
rosmmsg md5	Display the msg/srv md5 sum.
rosmmsg package	List all the msg/srv in a package.
rosmmsg packages	List all packages containing the msg/srv.

Examples:

```
Display the Pose msg:
$ rosmmsg show Pose
List the messages in the nav_msgs package:
$ rosmmsg package nav_msgs
List the packages using sensor_msgs/CameraInfo:
$ rosmmsg packages sensor_msgs/CameraInfo
```

Logging Tools

roscd

A set of tools for recording and playing back of ROS topics.

Commands:

roscd record	Record a bag file with specified topics.
roscd play	Play content of one or more bag files.
roscd compress	Compress one or more bag files.
roscd decompress	Decompress one or more bag files.
roscd filter	Filter the contents of the bag.

Examples:

```
Record select topics:
$ roscd record topic1 topic2
Replay all messages without waiting:
$ roscd play -a demo.log.bag
Replay several bag files at once:
$ roscd play demo1.bag demo2.bag
```

tf_echo

A tool that prints the information about a particular transformation between a source_frame and a target_frame.
Usage:

```
$ roscd tf tf_echo <source_frame> <target_frame>
```

Examples:

```
To echo the transform between /map and /odom:
$ roscd tf tf_echo /map /odom
```

