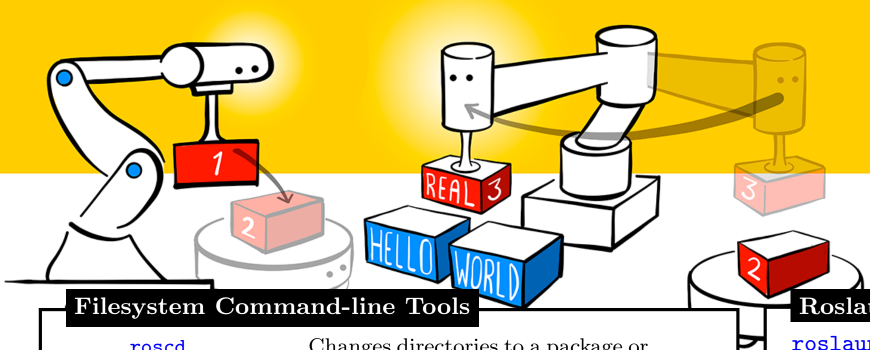

Hello (real) World with ROS Cheat Sheet

Filesystem Command-line Tools

roscd	Changes directories to a package or stack
rosls	Lists package or stack information
roscreate-pkg	Creates a new ROS package
roswtf	Displays errors and warnings about a running ROS system or launch file

Usage:

```
$ roscd [package[/subdir]]
$ rosls [package[/subdir]]
$ roscreate-pkg [package]
$ roswtf
$ roswtf [file]
```

Roscore

roscore is a collection of nodes and programs that are pre-requisites of a ROS-based system. You must have a roscore running in order for ROS nodes to communicate.

roscore is currently defined as:

master
parameter server
rosout

Usage:

```
$ roscore
```

Rosrun

rosrun allows you to run an executable in an arbitrary package without having to cd (or roscd) there first.

Usage:

```
$ rosrun package executable
```

Example - Run turtlesim:

```
$ rosrun turtlesim turtlesim_node
```

Roslaunch

roslaunch starts ROS nodes both locally and remotely via SSH, as well as setting parameters on the parameter server.

Example - Launch the turtlebot simulation:

```
$ roslaunch turtlebot_gazebo turtlebot_world.launch
```

Rosnode

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

Commands:

rosnode ping	Test connectivity to node
rosnode list	List active nodes
rosnode info	Print information about a node
rosnode kill	Kills a running node

Rostopic

rostopic is a tool for displaying debug information about ROS topics, including publishers, subscribers, publishing rate, and messages.

Commands:

rostopic echo	Print messages to screen
rostopic hz	Display publishing rate of topic
rostopic list	List active topics
rostopic pub	Publish data to topic
rostopic type	Print topic type
rostopic find	Find topics by type

Rosparam

rosparam is a tool for getting and setting ROS parameters on the parameter server, using YAML-encoded files.

Commands:

rosparam set	Set a parameter
rosparam get	Get a parameter
rosparam list	List parameter names

Rosservice

rosservice is a tool for listing and querying ROS services.

Commands:

rosservice list	Print a list of active services
rosservice node	Print the name of the 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 find	Find services by service type

tf Command-line Tools

tf_echo is a tool that prints information about a particular transformation between a source_frame and a target_frame.

Usage:

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

Example - Echo transform between /map and /odom:

```
$ rosrun tf tf_echo /map /odom
```

view_frames is a tool for visualizing the full tree of coordinate transforms.

Usage:

```
$ rosrun tf view_frames
$ evince frames.pdf
```