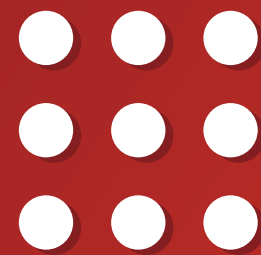


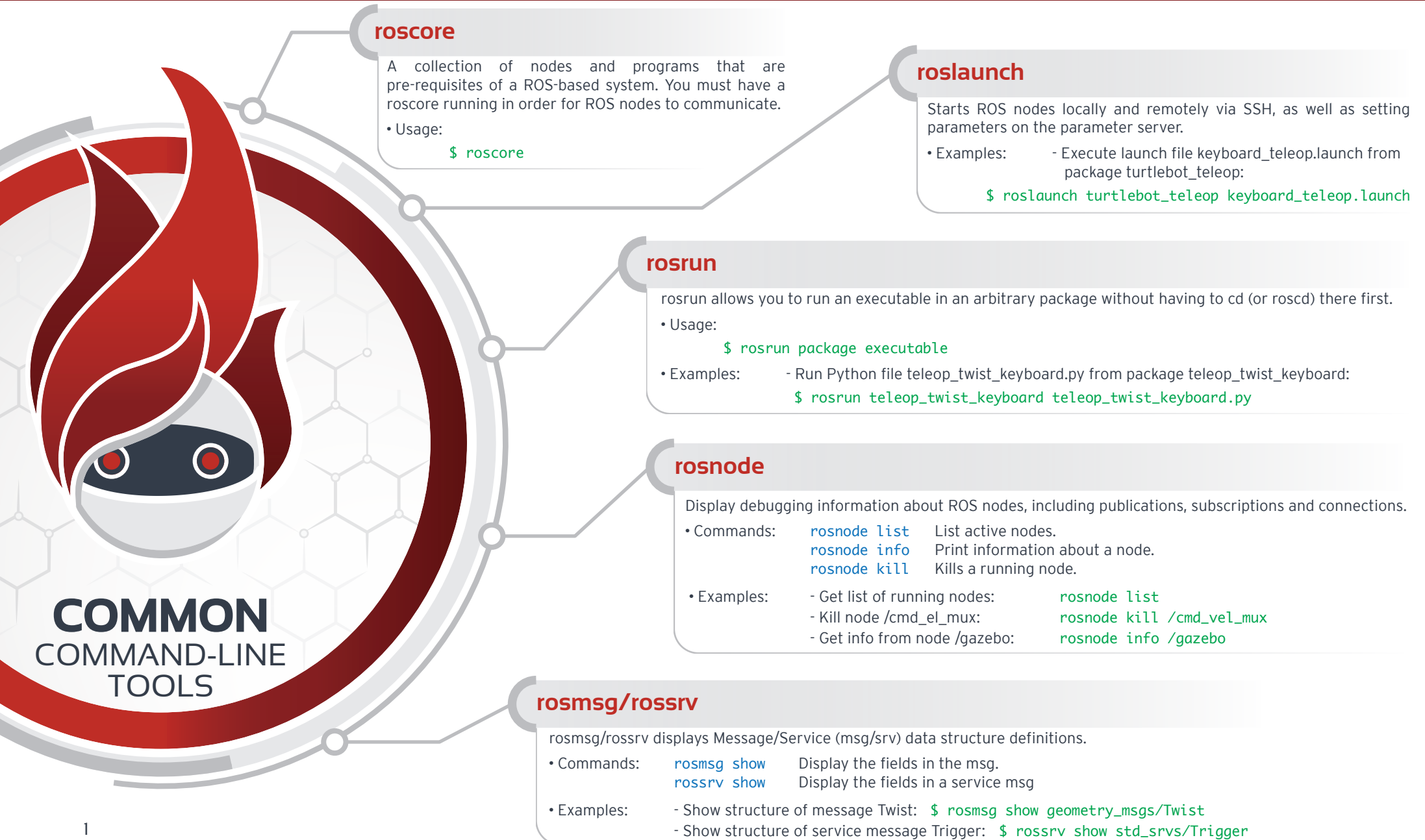


robotignite
ACADEMY



ROS

IN 5 PAGES





rostopic

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

- Commands:

<code>rostopic echo</code>	Print messages to screen.
<code>rostopic info</code>	Print information about a topics.
<code>rostopic list</code>	Print information about active topics.
<code>rostopic pub</code>	Publish data to topic.

- Examples:

- Get list of running topics: `rostopic list`
- Get info from topic /cmd_vel: `rostopic info /cmd_vel`
- Listen to topic /cmd_vel: `rostopic echo /cmd_vel`
- Publish a Twist message into topic /cmd_vel: `rostopic pub /cmd_vel geometry_msgs/ Twist "linear: x: 0.0 y: 0.0 z: 0.0 angular: x: 0.0 y: 0.0 z: 0.0"`

rosparam

A tool for getting and setting ROS parameters on the parameter server using YAML-encoded files.

- Commands:

<code>rosparam set</code>	Set a parameter.
<code>rosparam get</code>	Get a parameter.
<code>rosparam delete</code>	Delete a parameter.
<code>rosparam list</code>	List parameter names.

- Examples:

- List parameters loaded on the ROS Param Server: `rosparam list`
- Get value of parameter /camera/imager_rate: `rosparam get /camera/imager_rate`
- Set parameter /camera/imager_rate to 3.0: `rosparam set /camera/imager_rate 3.0`

rosservice

A tool for listing and querying ROS services.

- Commands:

<code>rosservice list</code>	Print information about active services.
<code>rosservice type</code>	Print the service type.
<code>rosservice info</code>	Get information from a service
<code>rosservice call</code>	Call the service with the given args.

- Examples:

- Get list of running services: `rosservice list`
- Get service message used by the service /gazebo/delete_model: `rosservice type /gazebo/delete_model`
- Get information from the service /gazebo/delete_model: `rosservice info /gazebo/delete_model`
- Call service /gazebo/delete_model and provide argument model_name: `rosservice call /gazebo/delete_model "model_name:'bowl'"`



Filesystem Command-line Tools

- Commands:
 - `rospack` A tool inspecting packages.
 - `roscd` Changes directories to a package or stack.
 - `roswtf` Displays a errors and warnings about a running ROS system or launch file.
 - `catkin_create_pkg` A tool for creating a new packages.
- Usage:
 - `$ rospack find [package]`
 - `$ roscd [package [/subdir]]`
 - `$ roswtf or roswtf [file]`
 - `$ catkin_create_pkg [name_of_pkg][dependencies]`
- Examples:
 - Find pack of package turtlebot_gazebo: `$ rospack find turtlebot_gazebo`
 - Go to package turtlebot_gazebo: `$ roscd turtlebot_gazebo`
 - Check for errors in keyboard_teleop.launch file: `$ roswtf keyboard_teleop.launch`



rosvag

This is a set of tools for recording from and playing back to ROS topics. It is intended to be high performance and avoids deserialization and reserialization of the messages.

rosvag record will generate a ".bag" file (so named for historical reasons) with the contents of all topics that you pass to it.

- Examples:

- Record all topics: `$ rosvag record -a`
- Record select topics: `$ rosvag record topic1 topic2`

rosvag play will take the contents of one or more bag file, and play them back in a time-synchronized fashion.

- Examples:

- Replay test.bag file: `$ rosvag play test.bag`
- Replay several bag files at once: `$ rosvag info test.bag`

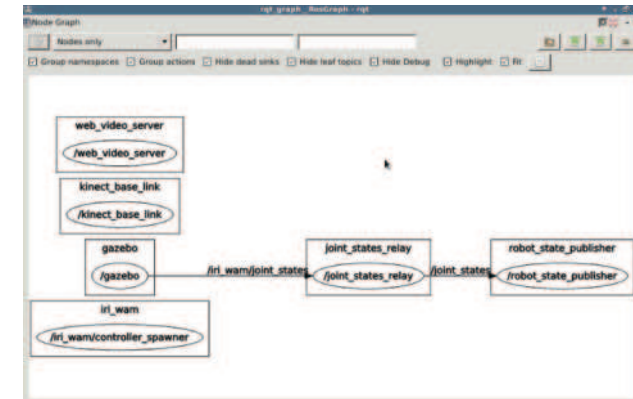


rx_graph

Displays a graph of the ROS nodes that are currently running, as well as the ROS topics that connect them.

• Usage:

```
$ rqt_graph
```



view_frames

A tool for visualizing the full tree of coordinate transforms.

• Usage:

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

rx_plot

A tool for plotting data from one or more ROS topic fields.

• Usage:

```
$ rqt_plot
```

