ROS 2 Cheats Sheet

All ROS 2 CLI tools start with the prefix 'ros2' followed by a verb and (possibly) a sub-verb. For any tool, the documentation is accessible with,

```
$ ros2 verb - -help
```

and similarly for sub-verb documentation,

```
$ ros2 verb sub_verb -h
```

action Also allows to manually send a goal and displays debugging information about action.
Sub-commands:

info	Output information about an action.
list	Output a list of action names.
send_goal	Send an action goal.
show	Output the action definition.

bag Allows to record/play topics to/from a rosbag.
Sub-commands:

info	Output information of a bag.
play	Play a bag.
record	Record a bag.

Examples:

```
$ ros2 info bag_name
$ ros2 play bag_name
$ ros2 record -a
```

component Various component related sub-commands.
Sub-commands:

list	Output a list of running containers and components.
load	Load a component into a container node.
standalone	Run a component into its own standalone container node.
types	Output a list of components registered in the ament index.
unload	Unload a component from a container node.

daemon Various daemon related sub-commands.
Sub-commands:

start	Start the daemon if it isn't running.
status	Output the status of the daemon.
stop	Stop the daemon if it is running

extension_points List extension points.

extensions List extensions.

launch Allows to run a launch file in an arbitrary package without to cd there first.
Usage:

```
$ ros2 launch <package> <launch-file>
```

Example:

```
$ ros2 launch demo_nodes_cpp add_two_ints.launch.py
```

lifecycle Various lifecycle related sub-commands.
get list nodes set Sub-commands:

get	Get lifecycle state for one or more nodes.
list	Output a list of available transitions.
nodes	Output a list of nodes with lifecycle.
set	Trigger lifecycle state transition.

msg Displays debugging information about messages.
Sub-commands:

list	Output a list of message types.
package	Output a list of message types within a given package.
packages	Output a list of packages which contain messages.
show	Output the message definition.

Examples:

```
$ ros2 msg list
$ ros2 msg package std_msgs
$ ros2 msg packages
$ ros2 msg show geometry_msgs/msg/Pose
```

multicast Various multicast related sub-commands.
Sub-commands:

receive	Receive a single UDP multicast packet.
send	Send a single UDP multicast packet.

node Displays debugging information about nodes.
Sub-commands:

info	Output information about a node.
list	Output a list of available nodes.

Examples:

```
$ ros2 node info /talker
$ ros2 node list
```

param Allows to manipulate parameters.
Sub-commands:

delete	Delete parameter.
get	Get parameter.
list	Output a list of available parameters.
set	Set parameter

Examples:

```
$ ros2 param delete \talker \use_sim_time
$ ros2 param get \talker \use_sim_time
$ ros2 param list
$ ros2 param set \talker \use_sim_time false
```

pkg
Sub-commands:

create	Create a new ROS2 package.
executables	Output a list of package specific executables.
list	Output a list of available packages.
prefix	Output the prefix path of a package.

Examples:

```
$ ros2 pkg executables demo_nodes_cpp
$ ros2 pkg list
$ ros2 pkg prefix std_msgs
```

run Allows to run an executable in an arbitrary package without having to cd there first.
Usage:

```
$ ros2 run <package> <executable>
```

Example:

```
$ ros2 run demo_node_cpp talker
```

security Various security related sub-commands.
Sub-commands:

create_key	Create key.
create_permission	Create keystore.
generate_artifacts	Create permission.
list_keys	Distribute key.
create_keystore	Generate keys and permission files from a list of identities and policy files.
distribute_key	Generate XML policy file from ROS graph data.
generate_policy	List keys.

service Allows to manually call a service and displays de-

bugging information about services.
Sub-commands:

call	Call a service.
find	Output a list of services of a given type.
list	Output a list of service names.
type	Output service's type.

Examples:

```
    $ ros2 service call \
/add_two_ints example_interfaces/AddTwoInts "a: 1, b: 2"
    $ ros2 service find rcl_interfaces/srv/ListParameters
    $ ros2 service list
    $ ros2 service type /talker/describe_parameters
```

---

srv
Sub-commands:
- list
- package
- packages
- show

---

test

---

topic A tool for displaying debug information about ROS topics, including publishers, subscribers, publishing rate, and messages.
Sub-commands:

bw	Display bandwidth used by topic.
delay	Display delay of topic from timestamp in header.
echo	Output messages of a given topic to screen.
find	Find topics of a given type type.
hz	Display publishing rate of topic.
info	Output information about a given topic.
list	Output list of active topics.
pub	Publish data to a topic.
type	Output topic's type.

Examples:

```
    $ ros2 topic bw
    $ ros2 topic echo
    $ ros2 topic hz
```