

ROS 2 Cheat Sheet

Command Line Interface

All ROS 2 CLI tools start with the prefix 'ros2' followed by a command, a verb and (possibly) positional/optional arguments.

For any tool, the documentation is accessible with,

```
$ ros2 command --help
```

and similarly for verb documentation,

```
$ ros2 command verb -h
```

Similarly, auto-completion is available for all commands/verbs and most positional/optional arguments. E.g.,

```
$ ros2 command [tab][tab]
```

Some of the examples below rely on:

[ROS 2 demos package](#).

action Allows to manually send a goal and displays debugging information about actions.

Verbs:

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

Examples:

```
$ ros2 action info /fibonacci
$ ros2 action list
$ ros2 action send_goal /fibonacci \
  action_tutorials/action/Fibonacci "order: 5"
$ ros2 action show action_tutorials/action/Fibonacci
```

bag Allows to record/play topics to/from a rosbag.

Verbs:

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
```

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
```

msg Displays debugging information about messages.

Verbs:

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
```

node Displays debugging information about nodes.

Verbs:

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.

Verbs:

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 Create a ros2 package or output package(s)-related information.

Verbs:

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_nodes_cpp talker
```

service Allows to manually call a service and displays debugging information about services.

Verbs:

call Call a service.

list Output a list of service names.

Examples:

```
$ ros2 service call /add_two_ints \
  example_interfaces/AddTwoInts "{a: 1, b: 2}"
$ ros2 service list
```

srv Various srv related verbs.

Verbs:

list Output a list of available service types.

package Output a list of available service types within one package.

packages Output a list of packages which contain services.

show Output the service definition.

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

Verbs:

bw Display bandwidth used by topic.

delay Display delay of topic from timestamp in header.

echo Output messages of a given topic to screen.

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.

Examples:

```
$ ros2 topic bw /chatter
```

```
$ ros2 topic echo /chatter
```

```
$ ros2 topic hz /chatter
```

```
$ ros2 topic info /chatter
```

```
$ ros2 topic list
```

```
$ ros2 topic pub /chatter std_msgs/msg/String \
  'data: Hello ROS 2 world'
```

```
$ ros2 topic type /rosout
```