ROS 2 Cheats 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

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.

Examples:

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

bag

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

Verbs:

convert	Write out a new bag with different settings.
info	Output information of a bag.
list	Output list of available plugins.
play	Play a bag.
record	Record a bag.
reindex	Reconstruct metadata file for a bag.

Examples:

```
$ ros2 bag convert --input <bag-name> storage_id \
  --output-options <settings-file-name>
$ ros2 bag info <bag-name>
$ ros2 bag play <bag-name>
$ ros2 bag record -a
$ ros2 bag reindex <bag-dir>
```

component

component Various component related verbs.

Verbs:

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.

Examples:

```
$ ros2 component list
$ ros2 component load /ComponentManager \
 composition composition::Talker
$ ros2 component types
$ ros2 component unload /ComponentManager 1
```

daemon

daemon Various daemon related verbs.

Verbs:

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

doctor

doctor Check ROS setup and other potential issues

Alias: **wtf** (where’s the fire).

Arguments:

--report/-r	Output report of all checks.
--report-fail/-rf	Output report of failed checks only.
--include-warning/-iw	Include warnings as failed checks.

Examples:

```
$ ros2 doctor
$ ros2 doctor --report
$ ros2 doctor --report-fail
$ ros2 doctor --include-warning
$ ros2 doctor --include-warning --report-fail
```

or similarly,

```
$ ros2 wtf
```

extension_points

extension_points List extension points.

extensions

extensions List extensions.

interface

interface Various ROS interfaces (actions/topics/services)-related verbs. Interface type can be filtered with either of the following option, ‘--only-actions’, ‘--only-msgs’, ‘--only-srvs’.

Verbs:

list	List all interface types available.
package	Output a list of available interface types within one package.
packages	Output a list of packages that provide interfaces.
proto	Print the prototype (body) of an interfaces.
show	Output the interface definition.

Examples:

```
$ ros2 interface list
$ ros2 interface package std_msgs
$ ros2 interface packages --only-msgs
$ ros2 interface proto example_interfaces/srv/AddTwoInts
$ ros2 interface show geometry_msgs/msg/Pose
```

launch

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

lifecycle Various lifecycle related verbs.

Verbs:

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.

multicast

multicast Various multicast related verbs.

Verbs:

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

node

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.
describe	Show descriptive information about declared parameters.
dump	Dump the parameters of a given node in YAML format, either in terminal or in a file.
get	Get parameter.
list	Output a list of available parameters.
load	Load parameter file for a node.
set	Set parameter

Examples:

```
$ ros2 param delete /talker <param-name>
$ ros2 param get /talker <param-name>
$ ros2 param list
$ ros2 param set /talker <param-name> <param-value>
```

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

create	Create a new ROS 2 package.
executables	Output a list of package specific executables.
list	Output a list of available packages.
prefix	Output the prefix path of a package.
xml	Output the XML of the package manifest or a specific tag.

Examples:

```
$ ros2 pkg executables demo_nodes_cpp
$ ros2 pkg list
$ ros2 pkg prefix std_msgs
$ ros2 pkg xml -t version
```

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 verbs.
Verbs:

create_enclave	Create enclave.
create_keystore	Create keystore.
create_permission	Create permission.
generate_artifacts	Generate keys and permission files from a list of identities and policy files.
generate_policy	Generate XML policy file from ROS graph data.
list_enclaves	List enclaves in keystore.

Examples (see sros2 package):

```
$ ros2 security create_enclave demo_keys /talker
$ ros2 security create_permission demo_keys /talker \
 policies/sample_policy.xml
$ ros2 security generate_artifacts
$ ros2 security create_keystore demo_keys
```

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

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
```

topic A tool for displaying debug information about ROS 2 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.
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 /chatter
$ ros2 topic echo /chatter
$ ros2 topic find rcl_interfaces/msg/Log
$ 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
```