

# ROS2 cheatsheet

---

## Introduction

\$ ros2 **command** **verb** **option**

---

## Executables

**run** run an executable in an arbitrary package

\$ ros2 run pkg exec

**launch** run a launch file in an arbitrary packages

\$ ros2 launch pkg file.launch.py

\$ ros2 launch pkg file.launch

\$ ros2 launch pkg file.launch.yaml

**component** manage components

\$ ros2 component **verb**

**list**

list of running containers and components

**load /ComponentManager** cnt ns::cmp cnt ns::cmp

load a component into a container node

**unload /ComponentManager** 1

unload a component from a container node

**standalone**

run a component into its own standalone container node

**types**

list of components registered in the ament index

---

## Package management

**pkg** create and manage ros2 package

\$ ros2 pkg **verb**

**create**

create a new ROS2 package

**executables** pkg

list of package specific executables

**list**

list of available packages

**prefix** pkg

prefix path of a package

**xml** -t version

information contained in the package xml manifest

**node** debugging information about nodes

\$ ros2 node **verb**

**info** node

information about a node

**list**

list of available nodes

---

## Interface management

**topic** topic interface

\$ ros2 topic **verb**

**list**

list of active topics

**info** topic

topic info

**echo** topic

echo of topic data

**hz** topic

topic data rate

**bw** topic

topic bandwidth

**delay** topic

topic delay from timestamp

**type** topic

topic's data type

**find** msgs

find topics of a given data type

**pub** msgs data

publish data to a topic

**service** service interface

\$ ros2 service **verb**

**list**

list of active service

**type** service

service's data type

**find** service

list of services of a given type

**call** srvs data

call a service

**action** action interface

\$ ros2 action **verb**

**list**

list of active action

**show** action

action definition

**info** action

information about an action

**send\_goal** act "data"

send action goal

**param** parameter interface

\$ ros2 param **verb**

**list**

list of available parameters

**describe** param

parameter info

**get** param

get parameter

**set** param data

set parameter

**delete** param

delete parameter

**dump** param data

dump parameter of a node in yaml, file or terminal

**interface** interface (actions/topics/services) commands

\$ ros2 interface **verb**

**list**

list all interface types available

**show** msg

interface definition

**proto** msg

prototype (body) of an interfaces

**package** pkg

list of available interface types within one package

**packages** --only-msgs

list of packages that

--only-srvs

provide interfaces

--only-actions

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

\$ ros2 bag **verb**

**info**

bag info

**record** bag\_name

record a bag

**play** bag\_name

play a bag

## Diagnostics

**lifecycle** Lifecycle CLI

\$ ros2 lifecycle **verb**

**list**

list of available transitions

**nodes**

list of nodes with lifecycle

**get** node

get node lifecycle state

**set** ??

trigger lifecycle state transition

**doctor** Check ROS setup and other potential issues

\$ ros2 doctor

-r report of all checks

-rd report of failed checks only

-iw report with warnings as failed checks

**test** Run ROS2 launch test

```
$ ros2 test
```

---

## Various

**daemon** Deamon CLI

```
$ ros2 daemon
```

**start** start the daemon if it isn't running

**status** status of the daemon

**stop** stop the daemon if it is running

**multicast** Various multicast related verbs.

Verbs:

**receive** Receive a single UDP multicast packet.

**send** Send a single UDP multicast packet.

**extension\_points** List extension points.

**extensions** List extensions.

**security** Various security related verbs.

Verbs:

**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.

Examples (see [sros2 package](#)):

```
$ ros2 security create_key demo_keys /talker
```

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

```
$ ros2 security generate_artifacts
```

```
$ ros2 security create_keystore demo_keys
```

```
$ colcon test-result
```

```
$ colcon version-check
```

```
$ colcon extension-point
```

```
$ colcon extensions
```

---

## ROS2 workspace

Creating workspace:

```
$ source /opt/ros/${ROS_DISTRO}/setup.bash
```

```
$ mkdir -p ~/ros2_ws/src
```

```
$ cd ~/ros2_ws/src
```

```
$ rosdep install -i --from-path src --rosdistro ROS_DISTRO
```

```
$ colcon build
```

```
$ source /opt/ros/ROS_DISTRO/setup.bash
```

```
$ source /opt/ros/ROS_DISTRO/setup.bash
```

---

## Building Tool

**colcon** run an executable in an arbitrary package

```
$ colcon build
```

```
$ colcon info
```

```
$ colcon list
```

```
$ colcon metadata
```

```
$ colcon test
```