ROS 2 CHEAT SHEET JAZZY

ROS.org

WORKSPACES

Create Workspace

```
mkdir -p colcon_ws/src
cd colcon_ws
colcon build
source install/setup.bash
```

Add Source Package to Workspace

```
cd /path/to/colcon_ws/src
git clone https://github.com/org/repo_name.git -b repo_
branch
```

Resolve Dependencies in Workspace

```
sudo rosdep init  # only once
cd /path/to/colcon_ws/
rosdep update
rosdep install --from-paths src --ignore-src -r -y
```

PACKAGES

Create a Package

CMake Package:

```
cd /path/to/colcon_ws/src
ros2 pkg create --build-type ament_cmake [--license
  your_license] package_name
```

Python Package:

```
cd /path/to/colcon_ws/src
ros2 pkg create --build-type ament_python [--license
  your_license] package_name
```

Package Files and Folders

package.xml	ROS package information	**package_name/**	Python3 source files
CMakeLists.txt	CMake package build instructions	**include/package_name/**	C++ header files
setup.cfg	Python setuptools configuration	**src/**	C++ source files
setup.py	Python package build instructions	**action/, 'msg/, 'srv/**	ROS action, message, and service definitions

CMakeLists.txt

Skeleton

```
cmake_minimum_required(VERSION 3.8)
project(package_name)
find_package(ament_cmake REQUIRED)
ament_package()
```

Actions, Message, and Services

Add all of your .action, .msg, and .srv files to CMakeLists.txt:

```
rosidl_generate_interfaces(
    ${PROJECT_NAME}
        action/my_action.action
        msg/my_message.msg
        srv/my_service.srv
)
```

Build Libraries and Executables

Each package can contain multiple executables or libraries:

```
add_executable(my_executable
  src/my_executable_source.cpp)

add_library(my_library src/my_library_source.cpp)
```

Installation

```
install(PROGRAMS my_executable
  DESTINATION lib/${PROJECT_NAME})
install(TARGETS my_library
  LIBRARY DESTINATION lib
  RUNTIME DESTINATION lib/${PROJECT_NAME})
install(DIRECTORY config launch
  DESTINATION share/${PROJECT_NAME})
```

Release Repo Packages

```
catkin_generate_changelog
# review & commit changelogs
catkin_prepare_release
bloom-release --track jazzy --ros-distro jazzy repo_name
```

setup.py

Skeleton

```
from setuptools import find_packages, setup
package_name = 'package_name'
setup(
    name=package_name,
    version='0.0.0',
    packages=find_packages(exclude=['test']),
    data_files=[
        ('share/ament_index/resource_index/packages',
            ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
    ],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='Maintainer Name',
    maintainer_email='my_email@my_provider.domain',
    description='Package description',
    license='your_license',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
        ],
    },
)
```

Installation

```
entry_points={
 'console_scripts': [
   'my_executable = my_executable.MyExecutableNode:main'
    ],
}

data_files = [
    ('share/ament_index/resource_index/packages',
        ['resource/' + package_name]),
    ('share/' + package_name, ['package.xml']),
    (os.path.join('share', package_name, 'config'), glob(
        os.path.join('config', '*.yaml'))),
    (os.path.join('share', package_name, 'launch'), glob(
        os.path.join('launch', '*.launch.py'))),
]
```

CLEARPATH ROBOTICS by ROCKWELL AUTOMATION

www.clearpathrobotics.com/ros-cheat-sheet

ROS 2 CHEAT SHEET JAZZY

ROS.org

ROS COMMAND-LINE TOOLS

Since ROS 2 launched, all ROS 2 terminal commands are of the format ros2 command [verb] [args...]

Setting the ROS_DOMAIN_ID

Because of its distributed nature, ROS 2 uses numbered domains to manage communication between nodes. Before running any ros2 commands, make sure your domain is set correctly.

```
export ROS_DOMAIN_ID=0
```

Domain IDs can be any integer from 0 to 232, though general convention is to avoid using any domain ID higher than 100 unless absolutely necessary.

If unset, the default domain ID is 0.

Interacting with Nodes

Start a Node	`ros2 run package_name executable_name [args] [--ros-args [-r original_topic:=remapped_topic] [-p parameter_name:=parameter_value]]`
Start Multiple Nodes with a Launch File	`ros2 launch [--debug] package_name launch_file.launch.py [launch arguments...]`
List Running Nodes	`ros2 node list`
Inspect a Node	`ros2 node info /fully/qualified/node_name`

Restart the ROS 2 Daemon

Running any ros2 command will automatically start the daemon if it isn't running already. You can stop the daemon by running

```
ros2 daemon stop
```

The daemon will automatically restart the next time you run a ROS 2 command.

Interacting with Topics

List Published Topics	`ros2 topic list`
Inspect a Topic	`ros2 topic info /fully/qualified/topic_name [--verbose]`
Echo a Topic	`ros2 topic echo /fully/qualified/topic_name`
Check the Publish Rate of a Topic	`ros2 topic hz /fully/qualified/topic_name`
Publish a Topic	ros2 topic pub /fully/qualified/topic_name package/msg/Type '{ JSON-formatted message payload goes here }' [-r HZ]

Interacting with Services

List Services	`ros2 service list`
Inspect a Service	`ros service info /fully/qualified/service_name`
Call a Service	`ros2 service call /fully/qualified/service_name package/srv/Type ['{ JSON-formatted service args here }']`

Interacting with Actions

List Actions	`ros2 actions list`
Inspect an Action	`ros2 action info /fully/qualified/action_name`
Send a Goal to an Action	`ros actions info /fully/qualified/action_name ['{ JSON-formatted action goal }']`

MORE RESOURCES

We have more resources to help you succeed with your projects and bring your robot vision to life.

Documentation

Visit our online documentation portal for step-by-step guides, software downloads, and support resources at docs.clearpathrobotics.com.

More information is also available at github.com/clearpathrobotics.

Robotics Content

Get inspired by robotics content on our blog, social media, and videos.

YouTube: youtube.com/@Clearpath
Blog: clearpathrobotics.com/blog/
LinkedIn: linkedin.com/company/clearpath-robotics/
Twitter/X: x.com/clearpathrobots

Components Store

Explore our Components Store for a wide selection of robotics components and accessories including LiDARS, cameras, manipulators, GNSS, IMUs and more: store.clearpathrobotics.com

Talk to a Human

To learn more about our robot platforms and integration services, connect with our team at info@clearpathrobotic.com.

For technical support, email us at support@clearpathrobotics.com or call 1-800-301-3863.

CLEARPATH ROBOTICS by ROCKWELL AUTOMATION

www.clearpathrobotics.com/ros-cheat-sheet