

## 4.ROS workspace

---

The ros workspace is a folder used to store ros function packages. It usually ends with ws. Let's create a space named ros\_ws as an example to explain how to create a ros workspace.

### 4.1 Create workspace folder

Take creating the ros\_ws space in the ~ directory as an example, and enter in the terminal,

```
cd
mkdir ros_ws
```

### 4.2 Create src to store function packages

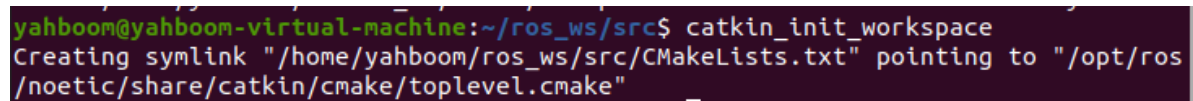
We create a src folder in the src directory of the workspace to store the function package created later, and enter it in the terminal.

```
cd ~/ros_ws
mkdir src
```

### 4.3 Initialize workspace

Terminal input,

```
cd ~/ros_ws/src
catkin_init_workspace
```

A terminal window screenshot showing the command 'catkin\_init\_workspace' being executed in the directory ~/ros\_ws/src. The output indicates that a symlink is being created for CMakeLists.txt, pointing to the system's catkin cmake directory.

```
yahboom@yahboom-virtual-machine:~/ros_ws/src$ catkin_init_workspace
Creating symlink "/home/yahboom/ros_ws/src/CMakeLists.txt" pointing to "/opt/ros/noetic/share/catkin/cmake/toplevel.cmake"
```

### 4.4 Compile workspace

Use the catkin\_make command to compile the functions in the entire workspace. You need to compile in the workspace directory,

```
cd ~/ros_ws
catkin_make
```

```

yahboom@yahboom-virtual-machine:~/ros_ws$ catkin_make
Base path: /home/yahboom/ros_ws
Source space: /home/yahboom/ros_ws/src
Build space: /home/yahboom/ros_ws/build
Devel space: /home/yahboom/ros_ws/devel
Install space: /home/yahboom/ros_ws/install
####
### Running command: "cmake /home/yahboom/ros_ws/src -DCATKIN_DEVEL_PREFIX=/home/yahboom/ros_ws/devel -DCMAKE_INSTALL_PREFIX=/home/yahboom/ros_ws/install -G Unix Makefiles" in "/home/yahboom/ros_ws/build"
####
-- The C compiler identification is GNU 9.4.0
-- The CXX compiler identification is GNU 9.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /home/yahboom/ros_ws/devel
-- Using CMAKE_PREFIX_PATH: /home/yahboom/orbbec_ws/devel;/home/yahboom/ArTrack_ws/devel;/opt/ros/noetic
-- This workspace overlays: /home/yahboom/orbbec_ws/devel;/home/yahboom/ArTrack_ws/devel;/opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.8.10", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Found PY_em: /usr/lib/python3/dist-packages/em.py
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/yahboom/ros_ws/build/test_results
-- Forcing gtest/gmock from source, though one was otherwise available.
-- Found gtest sources under '/usr/src/googletest': gtests will be built
-- Found gmock sources under '/usr/src/googletest': gmock will be built

```

## 4.5 Add workspace to environment variables

Add the workspace to the environment variable so that you can find the function package when you open the terminal. Otherwise, you need to source the workspace every time to find the function package and related programs. Enter in the terminal.

```
echo "source ~/ros_ws/devel/setup.bash" >> ~/.bashrc
```

Enter the following command to refresh the environment variables or reopen the terminal to take effect. Enter in the terminal,

```
source ~/.bashrc
```