

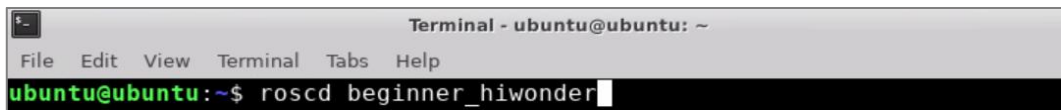
Lesson 10 The Definition and Use of Service Data

1. Customize Service Data

Note: Before customizing service data, the workspace and package need to be created first. The specific operation steps can be viewed in file “ROS Basic Lessons/Lesson 3 Create Workspace and Package”.

The specific operation steps to customize service data are as follow:

- 1) Open the terminal.
- 2) Enter “`roscd beginner_hiwonder`” command to go to the package directory and press “Enter”.

A terminal window titled "Terminal - ubuntu@ubuntu: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command prompt shows "ubuntu@ubuntu: ~\$ roscd beginner_hiwonder" with a cursor at the end.

Note: If there is a prompt “**No such package/stack ‘beginner_hiwonder’**” appears, it means that the package does not exist in the environment variable ROS_PACKAGE_PATH. The specific solution can be viewed in “ROS Basic Lessons/Lesson 3 Create Workspace and Package”. After the problem is solved, please repeat the current step.

- 3) Enter “`mkdir srv`” command and press “Enter”. Then create a new folder “srv” for storing text files.

A terminal window titled "Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The command prompt shows "ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder\$ mkdir srv" with a cursor at the end.

- 4) Enter “`vi Person.srv`” command to edit program, and then copy the following program. If want to modify, you can press “i”. After modifying, press “Esc” and enter “`:.wq`” to save and exit.

```
Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/srv
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder/srv$ vi Person.srv
```

```
string name

int8 age

int8 sex

int8 unknown = 0

int8 male = 1

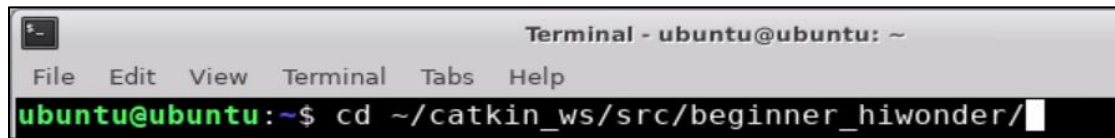
int8 female = 2

---

string result
```

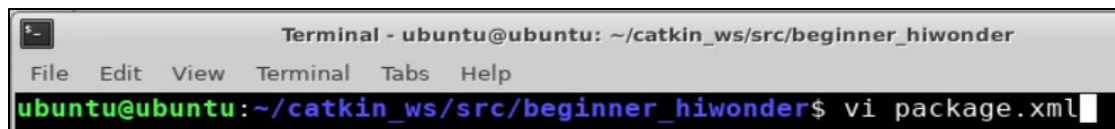
```
Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/srv
File Edit View Terminal Tabs Help
1 string name
2 int8 age
3 int8 sex
4
5 int8 unknown = 0
6 int8 male = 1
7 int8 female = 2
8
9 ---
10 string result
: wq
```

5) Enter “cd ~/catkin_ws/src/beginner_hiwonder/” command, and then press “Enter”.



```
Terminal - ubuntu@ubuntu: ~
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~$ cd ~/catkin_ws/src/beginner_hiwonder/
```

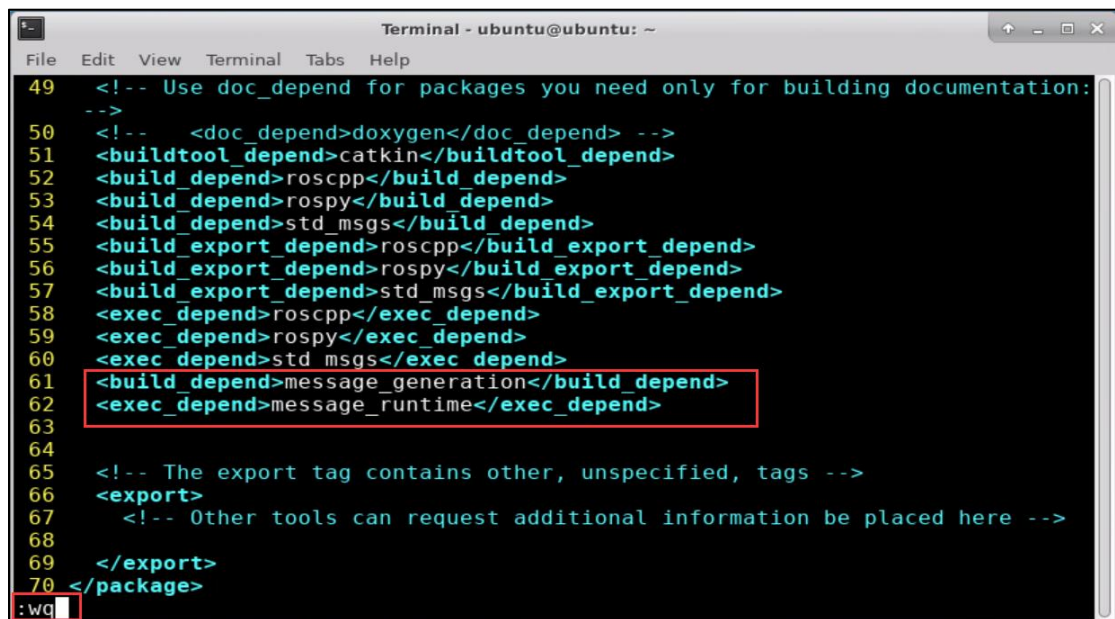
6) Enter “vi package.xml” command. Then copy the following program and add the package dependencies in the position shown in the figure below. If want to modify, you can press “i” again. After modifying, press “Esc” and enter “:wq” to save and exit.



```
Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder$ vi package.xml
```

```
<build_depend>message_generation</build_depend>

<exec_depend>message_runtime</exec_depend>
```



```
Terminal - ubuntu@ubuntu: ~
File Edit View Terminal Tabs Help
49 <!-- Use doc_depend for packages you need only for building documentation:
-->
50 <!-- <doc_depend>doxygen</doc_depend> -->
51 <buildtool_depend>catkin</buildtool_depend>
52 <build_depend>roscpp</build_depend>
53 <build_depend>rospy</build_depend>
54 <build_depend>std_msgs</build_depend>
55 <build_export_depend>roscpp</build_export_depend>
56 <build_export_depend>rospy</build_export_depend>
57 <build_export_depend>std_msgs</build_export_depend>
58 <exec_depend>roscpp</exec_depend>
59 <exec_depend>rospy</exec_depend>
60 <exec_depend>std_msgs</exec_depend>
61 <build_depend>message_generation</build_depend>
62 <exec_depend>message_runtime</exec_depend>
63
64
65 <!-- The export tag contains other, unspecified, tags -->
66 <export>
67 <!-- Other tools can request additional information be placed here -->
68
69 </export>
70 </package>
:wq
```

7) Enter “vi CMakeLists.txt” and press “i” to modify “CMakeLists.txt” file.



```
Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder$ vi CMakeLists.txt
```

8) Add the required compilation option “message_generation” in the position shown in the figure below.

```

6
7 ## Find catkin macros and libraries
8 ## if COMPONENTS list like find_package(catkin REQUIRED COMPONENTS xyz)
9 ## is used, also find other catkin packages
10 find_package(catkin REQUIRED COMPONENTS
11   roscpp
12   rospy
13   std_msgs
14   message_generation
15 )

```

9) Find the code shown in the figure below. Then uncomment the framed code and add the required compilation option “Person.srv”.

```

53 #   Message1.msg
54 #   Message2.msg
55 # )
56
57 ## Generate services in the 'srv' folder
58 add_service_files(
59   FILES
60   Person.srv
61 )
62
63 ## Generate actions in the 'action' folder
64 # add_action_files(

```

10) Find the code shown in the figure below. Then uncomment the code in red box and ensure that the required compilation options take effect.

```

67 #   Action2.action
68 # )
69
70 ## Generate added messages and services with any dependencies listed here
71 generate_messages(
72   DEPENDENCIES
73   std_msgs
74 )
75
76 #####
77 ## Declare ROS dynamic reconfigure parameters ##
78 #####

```

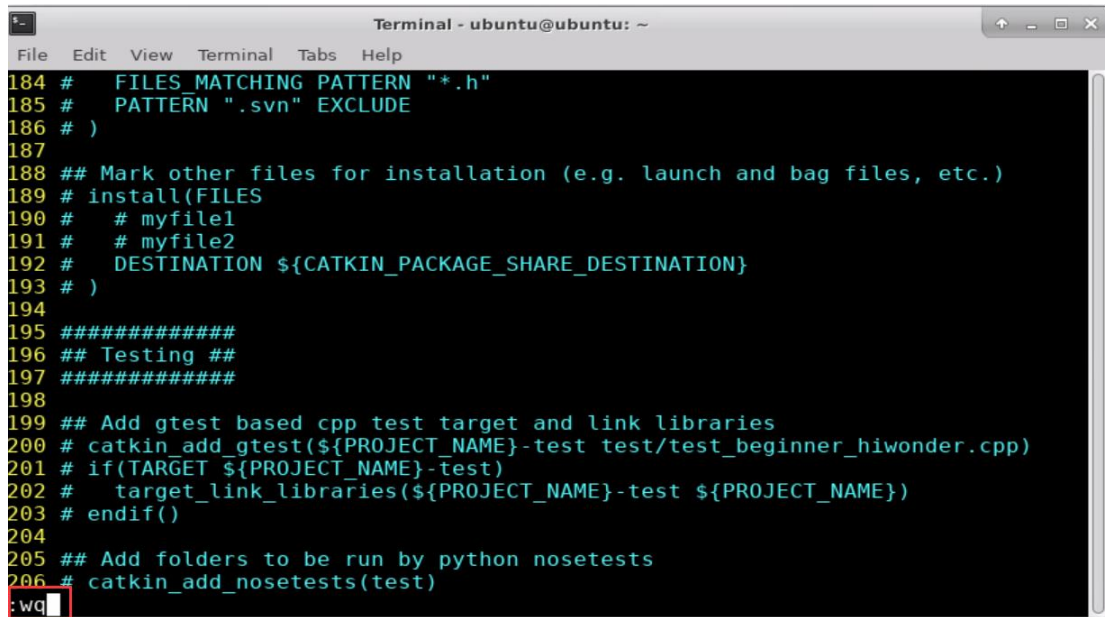
11) Find the code shown in the figure below. Then uncomment the code in red box and add the required compilation option “message_runtime”.

```

103 ## CATKIN_DEPENDS: catkin_packages dependent projects also need
104 ## DEPENDS: system dependencies of this project that dependent projects also
105   need
106 catkin_package(
107 #   INCLUDE_DIRS include
108 #   LIBRARIES beginner_hiwonder
109   CATKIN_DEPENDS roscpp rospy std_msgs message_runtime
110 #   DEPENDS system_lib
111 )

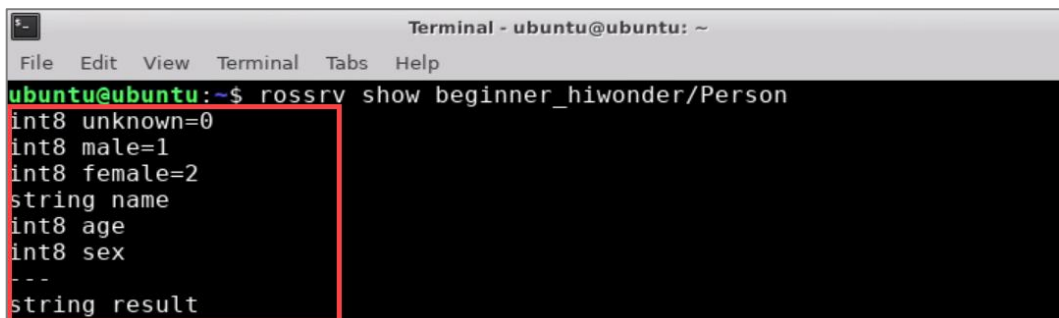
```

12) After modifying, press “Esc” and enter “:wq” to save and exit.

A terminal window titled "Terminal - ubuntu@ubuntu: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal displays CMakeLists.txt content with line numbers 184 to 206. Line 206, which contains the command "catkin_add_nosetests(test)", is highlighted with a red rectangular box.

```
184 # FILES_MATCHING PATTERN "*.h"
185 # PATTERN ".svn" EXCLUDE
186 # )
187
188 ## Mark other files for installation (e.g. launch and bag files, etc.)
189 # install(FILES
190 #   # myfile1
191 #   # myfile2
192 #   DESTINATION ${CATKIN_PACKAGE_SHARE_DESTINATION}
193 # )
194
195 #####
196 ## Testing ##
197 #####
198
199 ## Add gtest based cpp test target and link libraries
200 # catkin_add_gtest(${PROJECT_NAME}-test test/test_beginner_hiwonder.cpp)
201 # if(TARGET ${PROJECT_NAME}-test)
202 #   target_link_libraries(${PROJECT_NAME}-test ${PROJECT_NAME})
203 # endif()
204
205 ## Add folders to be run by python nosetests
206 # catkin_add_nosetests(test)
:wd
```

13) Enter the command “rosmv show beginner_hiwonder/Person” and press “Enter” to check whether the message written can be recognized by system. When the words shown in red box appear, it means that they are recognized successfully.

A terminal window titled "Terminal - ubuntu@ubuntu: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the command "rosmv show beginner_hiwonder/Person" and its output. The output lines, from "int8 unknown=0" to "string result", are enclosed in a red rectangular box.

```
ubuntu@ubuntu:~$ rosmv show beginner_hiwonder/Person
int8 unknown=0
int8 male=1
int8 female=2
string name
int8 age
int8 sex
---
string result
```

2. The Use of Service Data

2.1 Create Server and Client Data

- 1) Open the terminal.
- 2) Enter “cd catkin_ws/src/beginner_hiwonder/scripts/” command and press “Enter” to come to the folder “scripts” where Python scripts are stored.

```
Terminal - ubuntu@ubuntu: ~  
File Edit View Terminal Tabs Help  
ubuntu@ubuntu:~$ cd catkin_ws/src/beginner_hiwonder/scripts/
```

- 3) Enter “vi person_server.py” command to edit program, and then copy the following program. If want to modify, you can press “i”. After modifying, press “Esc” and enter “:wq” to save and exit.

```
Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts  
File Edit View Terminal Tabs Help  
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder/scripts$ vi person_server.py
```

```
#!/usr/bin/env python  
  
# -*- coding: utf-8 -*-  
  
# This routine will execute the /show_perso service with service data type  
beginner_hiwonder::Person  
  
import rospy  
  
from beginner_hiwonder.srv import Person, PersonResponse  
  
def personCallback(req):  
  
    # Display request message  
  
    rospy.loginfo("Person: name:%s age:%d sex:%d", req.name, req.age, req.sex)  
  
    # Data feedback  
  
    return PersonResponse("OK")
```



```
def person_server():

    # Initialize ROS node

    rospy.init_node('person_server')

    # Create a server named /show_person and register personCallback function

    s = rospy.Service('/show_person', Person, personCallback)

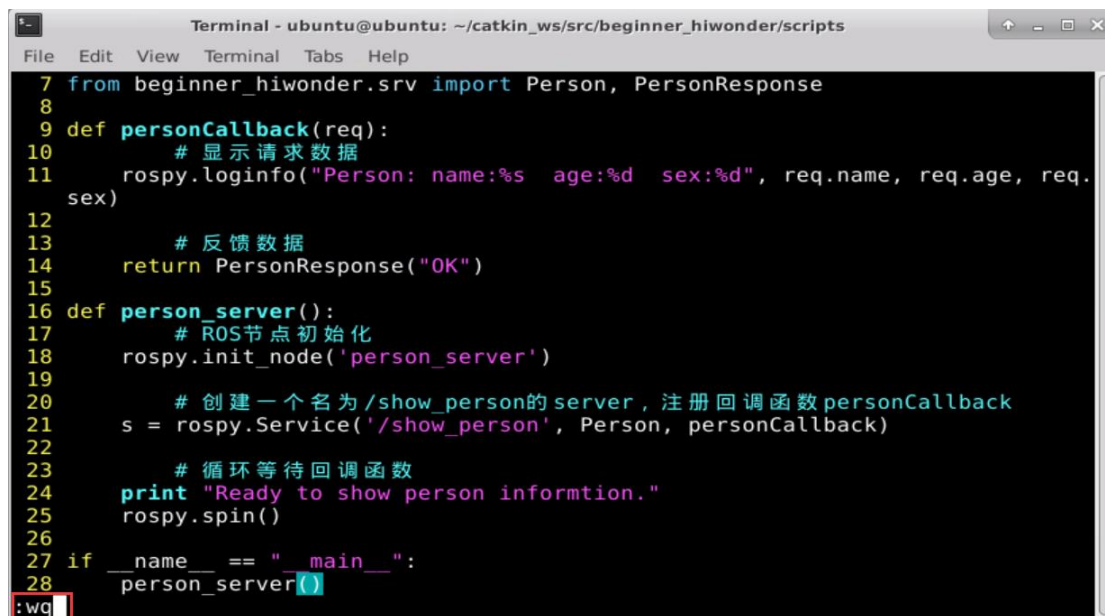
    # loop and wait callback function

    print "Ready to show person informtion."

    rospy.spin()

if __name__ == "__main__":

    person_server()
```

A screenshot of a terminal window titled "Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts". The terminal displays a Python script with line numbers 7 through 28. The script imports Person and PersonResponse from beginner_hiwonder.srv, defines a personCallback function that prints request data and returns an OK response, and a person_server function that initializes ROS, creates a service, and spins. The script is executed, and the prompt changes to :wq. The terminal output shows the script's execution steps: logging in, creating the service, and printing the ready message.

```
7 from beginner_hiwonder.srv import Person, PersonResponse
8
9 def personCallback(req):
10     # 显示请求数据
11     rospy.loginfo("Person: name:%s age:%d sex:%d", req.name, req.age, req.
sex)
12
13     # 反馈数据
14     return PersonResponse("OK")
15
16 def person_server():
17     # ROS节点初始化
18     rospy.init_node('person_server')
19
20     # 创建一个名为/show_person的server, 注册回调函数personCallback
21     s = rospy.Service('/show_person', Person, personCallback)
22
23     # 循环等待回调函数
24     print "Ready to show person informtion."
25     rospy.spin()
26
27 if __name__ == "__main__":
28     person_server()
:wq
```

- 4) Enter “vi person_subscriber.py” to edit program, and copy the following program. If want to modify, you can press “i”. After modifying, press “Esc”

and enter “:wq” to save and exit.



```
Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder/scripts$ vi person_client.py
```

```
#!/usr/bin/env python

# -*- coding: utf-8 -*-

# This routine will request /show_person service with the service data type
beginner_hiwonder::Person

import sys

import rospy

from beginner_hiwonder.srv import Person, PersonRequest

def person_client():

    # Initialize ROS node

    rospy.init_node('person_client')

    # After finding the /spawn service, create a service client, and then connect service name
    /spawn

    rospy.wait_for_service('/show_person')

    try:

        person_client = rospy.ServiceProxy('/show_person', Person)

    # Request service call, input request data
```



```

        response = person_client("Tom", 20, PersonRequest.male)

        return response.result

    except rospy.ServiceException, e:

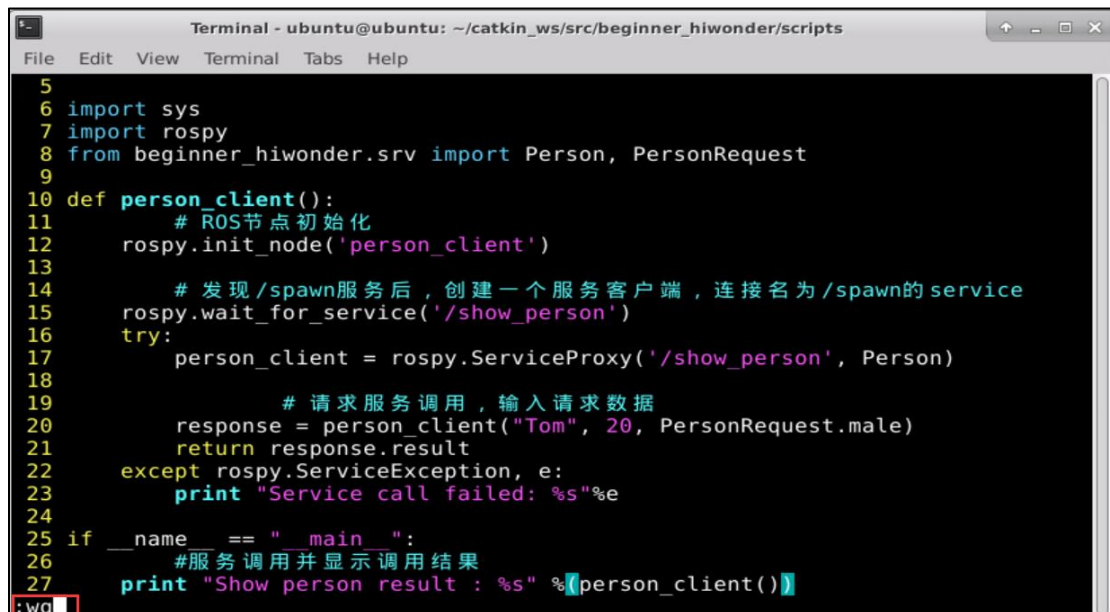
        print "Service call failed: %s"%e

if __name__ == "__main__":

    #Service call and display the call result

    print "Show person result : %s" %(person_client())

```



```

Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts
File Edit View Terminal Tabs Help
5
6 import sys
7 import rospy
8 from beginner_hiwonder.srv import Person, PersonRequest
9
10 def person_client():
11     # ROS节点初始化
12     rospy.init_node('person_client')
13
14     # 发现 /spawn服务后，创建一个服务客户端，连接名为 /spawn的 service
15     rospy.wait_for_service('/show_person')
16     try:
17         person_client = rospy.ServiceProxy('/show_person', Person)
18
19         # 请求服务调用，输入请求数据
20         response = person_client("Tom", 20, PersonRequest.male)
21         return response.result
22     except rospy.ServiceException, e:
23         print "Service call failed: %s"%e
24
25 if __name__ == "__main__":
26     #服务调用并显示调用结果
27     print "Show person result : %s" %(person_client())
:wg

```

- 5) Enter “chmod +x person_server.py” and “chmod +x person_client.py” command, and then press “Enter” to give the executable permission to the saved person_publisher.py.



```

Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder/scripts$ chmod +x person_server.py

```



```

Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder/scripts$ chmod +x person_client.py

```

2.2 Run Server and Client Nodes

- 1) Enter “cd ~/catkin_ws” command and press “Enter” to enter to catkin workspace.



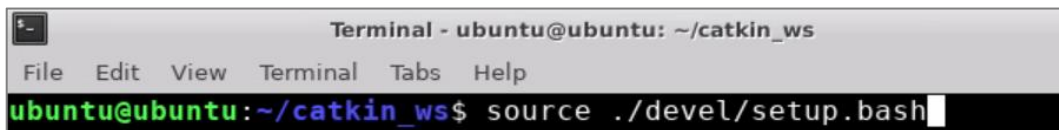
```
Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder/scripts$ cd ~/catkin_ws
```

- 2) Enter “catkin_make” command and press “Enter” to build all the packages in directory.




```
Terminal - ubuntu@ubuntu: ~/catkin_ws
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws$ catkin_make
```

- 3) Enter “source ./devel/setup.bash” command and press “Enter” to refresh the workspace environment.



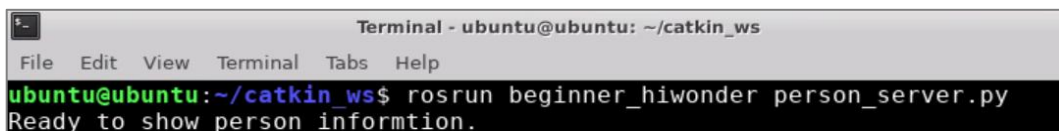
```
Terminal - ubuntu@ubuntu: ~/catkin_ws
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws$ source ./devel/setup.bash
```

- 4) Enter “roscore” command to start node manager.



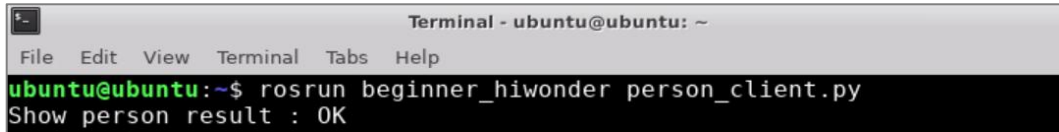
```
Terminal - ubuntu@ubuntu: ~/catkin_ws
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws$ roscore
```

- 5) Enter “roslaunch beginner_hiwonder person_publisher.py” command and press “Enter” to run publisher node. If want to stop running node, you can press “Ctrl+C”.



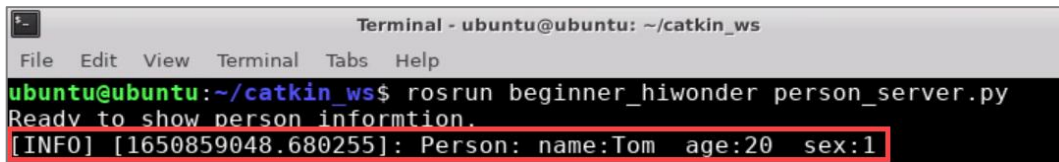
```
Terminal - ubuntu@ubuntu: ~/catkin_ws
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws$ roslaunch beginner_hiwonder person_server.py
Ready to show person information.
```

- 6) Open a new terminal. Enter “roslaunch beginner_hiwonder person_client.py” command and press “Enter” to run the client node.



```
Terminal - ubuntu@ubuntu: ~
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~$ rosrun beginner_hiwonder person_client.py
Show person result : OK
```

- 7) After running the client node, the terminal window for starting server node will print the content shown in the red box in the figure below.



```
Terminal - ubuntu@ubuntu: ~/catkin_ws
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws$ rosrun beginner_hiwonder person_server.py
Ready to show person information.
[INFO] [1650859048.680255]: Person: name:Tom age:20 sex:1
```