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”.

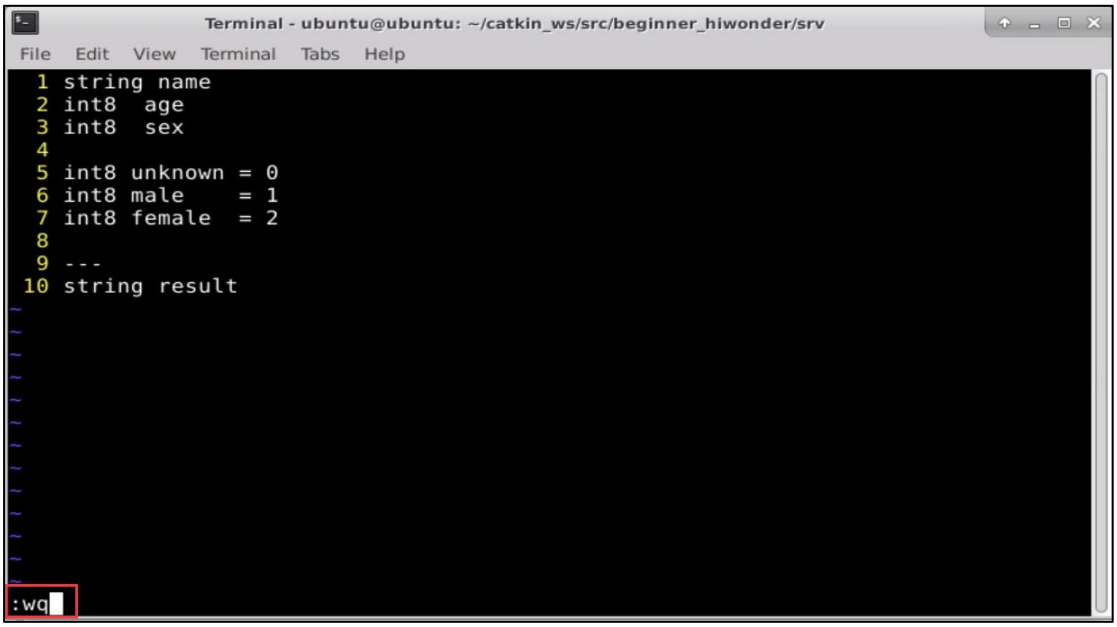
Note: If there is a prompt “**No such package/stack ‘beginner_hiwonder**” appears, it means that the package does not exist in the environment variable ROS_PACKAGE_PATH. The specific solution can be viewed in “ROS Basic Lessons/Lesson 3 Create Workspace and Package”. After the problem is solved, please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

---

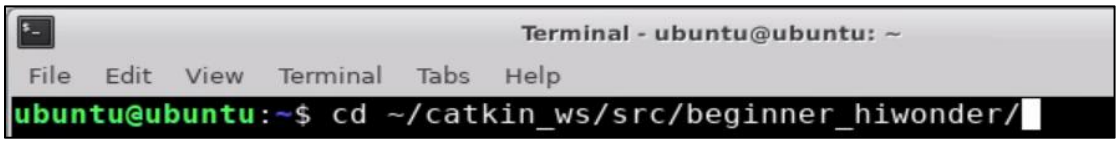
3) Enter “mkdir srv” command and press “Enter”. Then create a new folder “srv” for storing text files.

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.


```
string name

int8    age

int8    sex



int8 unknown = 0

int8 male       = 1

int8 female    = 2



---

string result
```

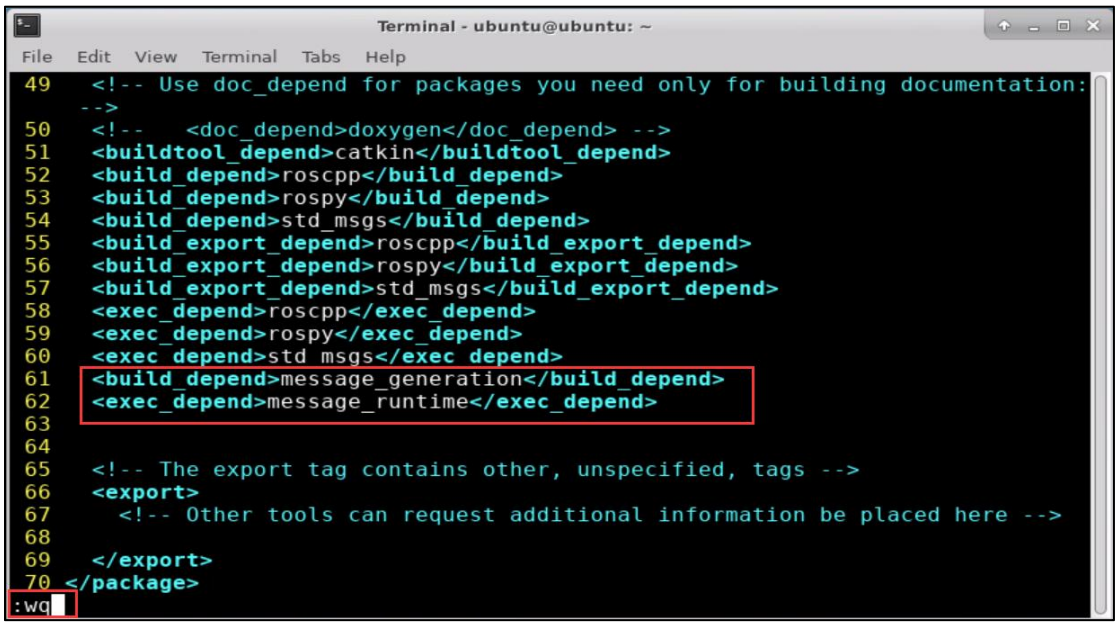


5) Enter “cd ~/catkin_ws/src/beginner_hiwonder/” command, and then press “Enter”.



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.


```
<build_depend>message_generation</build_depend>

<exec_depend>message_runtime</exec_depend>
```



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

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

```
 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(
```

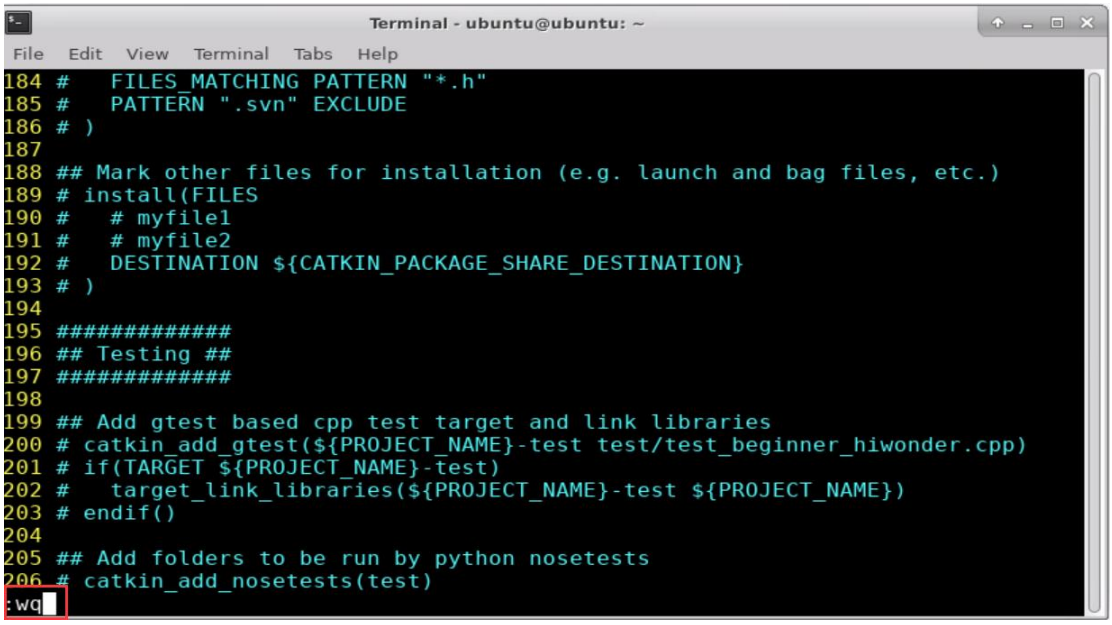
10) Find the code shown in the figure below. Then uncomment the code in red box and ensure that the required compilation options take effect.

```
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 ################################################
```

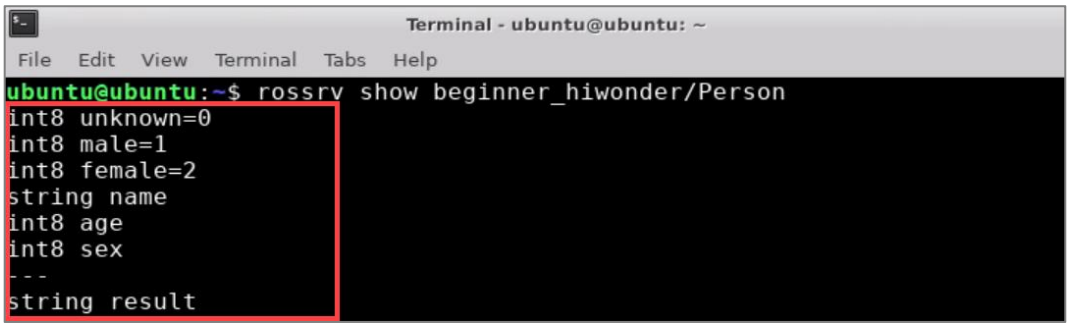
11) Find the code shown in the figure below. Then uncomment the code in red box and add the required compilation option “message_runtime”.

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

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



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



2. The Use of Service Data

2.1 Create Server and Client Data

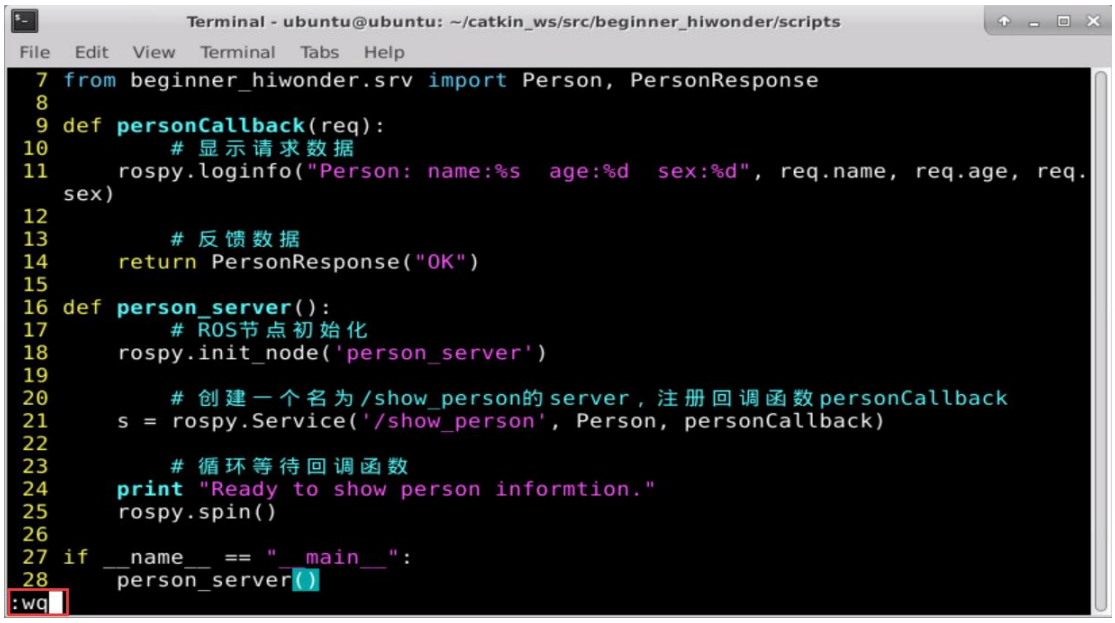
1) Open the terminal.

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

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.


```
#!/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()
```

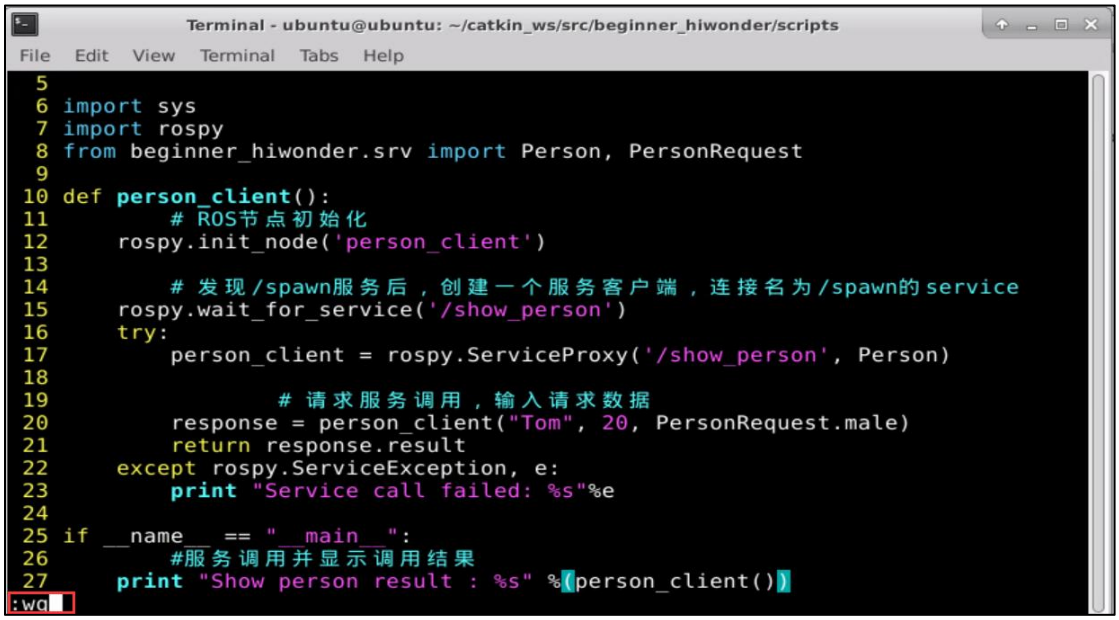


4) Enter “vi person_subscriber.py” to edit program, and copy the following program. If want to modify, you can press “i”. After modifying, press “Esc”

and enter ":wq" to save and exit.


```
#!/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())
```



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.

2.2 Run Server and Client Nodes

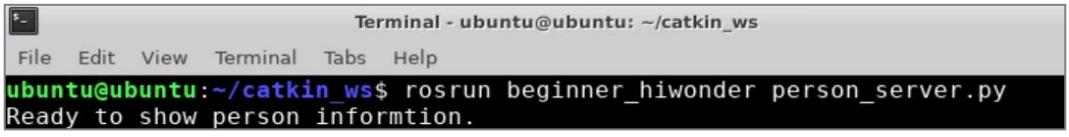
1) Enter “cd ~/catkin_ws” command and press “Enter” to enter to catkin workspace.

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

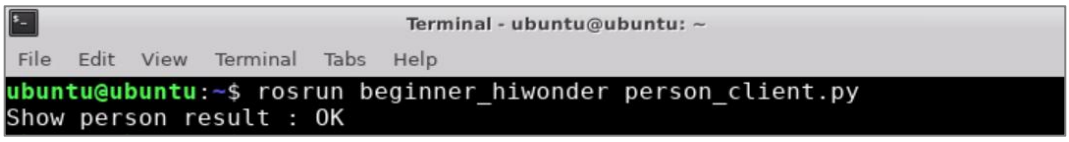
3) Enter “source ./devel/setup.bash” command and press “Enter” to refresh the workspace environment.

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

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



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



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.