6.Customized topic messages and usage

6.1 Customized topic messages

Switch to~/catkin_ ws/src/learning_ Create a new folder named msg under the topic feature pack directory to store custom topic messages.

6.1.1. Define msg files

Switch to the msg directory and create a new blank msg file, with the suffix "msg" indicating that it is an msg file. Here, we use Information.msg as an example to copy the following code into the newly created msg file.

```
string company
string city
```

6.1.2. Add feature package dependencies in package.xml

```
<build_depend>message_generation</build_depend>
<exec_depend>message_runtime</exec_depend>
```

6.1.3. Add compilation options in CMakeLists.txt

```
add_message_files(FILES Information.msg)
generate_messages(DEPENDENCIES std_msgs)
```

6.1.4 Compile and generate language related files

```
cd ~/catkin_ws
catkin_make
```

6.1.5. C++Language Implementation

1. Switch to~/catkin_ ws/src/learning_ Under topic/src, create two new cpp files named Information_ Publisher.cpp and Information_ Subscriber.cpp, copy the following code into it separately,Information_ publisher.cpp

```
#include <ros/ros.h>
#include "learning_topic/Information.h"

int main(int argc, char **argv)
{
    // ROS node initialization
    ros::init(argc, argv, "company_Information_publisher");

    // Create node handle
    ros::NodeHandle n;
```

```
    // Create a Publisher to publish a topic named/company info, with a message
type of learning_topic:: Person and a queue length of 10
    ros::Publisher Information_pub = n.advertise<learning_topic::Information>
("/company_info", 10);

    // Set the frequency of the loop
    ros::Rate loop_rate(1);

    int count = 0;
    while (ros::ok())
    {
        // Initialize learning_topic:: Information type message
        learning_topic::Information info_msg;
        info_msg.company = "Yahboom";
        info_msg.city = "Shenzhen";

        // Publish Message
        Information_pub.publish(info_msg);

        ROS_INFO("Information: company:%s  city:%s ",
                  info_msg.company.c_str(), info_msg.city.c_str());


        loop_rate.sleep();// Delay according to the cycle frequency
    }

    return 0;
}
```

Information_subscriber.cpp

```
#include <ros/ros.h>
#include "learning_topic/Information.h"

// After receiving the subscription message, it will enter the message callback
function to process the data
void CompanyInfoCallback(const learning_topic::Information::ConstPtr& msg)
{
    // Print the received message
    ROS_INFO("This is: %s  in %s", msg->company.c_str(), msg->city.c_str());
}

int main(int argc, char **argv)
{

    ros::init(argc, argv, "company_Information_subscriber");//Initialize ROS
nodes


    ros::NodeHandle n;// This is creating a node handle

    // Create a subscriber, subscribe to the topic/company info, and register the
callback function Company Infocallback
```

```
    ros::Subscriber person_info_sub = n.subscribe("/company_info", 10,
CompanyInfoCallback);


    ros::spin();// Loop waiting callback function

    return 0;
}
```

2. Modify the CMakeLists.txt file

```
add_executable(Information_publisher src/Information_publisher.cpp)
target_link_libraries(Information_publisher ${catkin_LIBRARIES})
add_dependencies(Information_publisher ${PROJECT_NAME}_generate_messages_cpp)

add_executable(Information_subscriber src/Information_subscriber.cpp)
target_link_libraries(Information_subscriber ${catkin_LIBRARIES})
add_dependencies(Information_subscriber ${PROJECT_NAME}_generate_messages_cpp)
```

3. Core part

The implementation process here is the same as before, with the main difference being the introduction of header files and the use of custom message files:The import header file is

```
#include "learning_topic/Information.h"
```

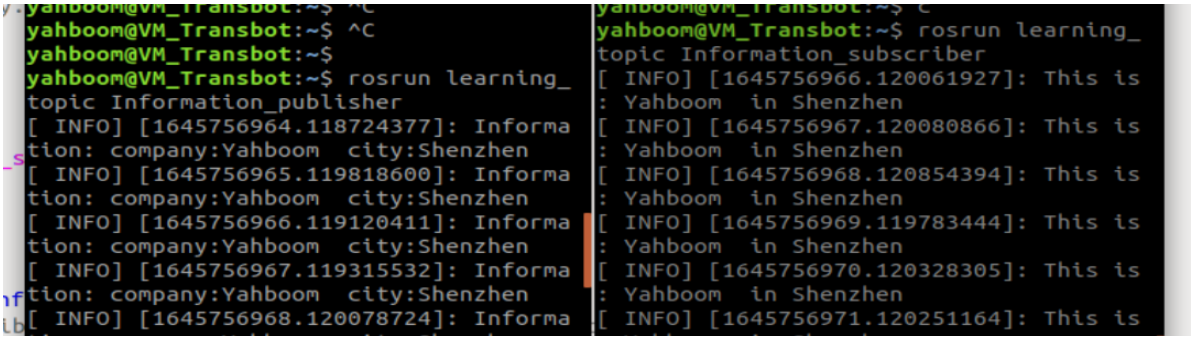
Front learning_ Topic is the name of the feature pack, followed by Information. h, which is the header file name generated by the previously created msg fileUsing a custom message file is

```
learning_topic::Information info_msg;
info_msg.company = "Yahboom";
info_msg.city = "Shenzhen";
void CompanyInfoCallback(const learning_topic::Information::ConstPtr& msg)
```

4. Run program

```
roscore
rosrun learning_topic Information_publisher
rosrun learning_topic Information_subscriber
```

5. Run screenshot



6. Program Description

Information_ As a publisher, publisher continuously posts messages to the topic of "/company_info" and prints the published messages; As a subscriber, Information_ The subscriber also continuously receives the content of the topic "/company_info" and prints it out in the callback function.

6.1.6 Python Language Implementation

1. Switch to~/catkin_ ws/src/learning_ Under topic/script, create two new py files and name them Information_ Publisher.py and Information_ Subscriber.py, copy the following code into it separately,Information_ publisher.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-


import rospy

from learning_topic.msg import Information #Import custom msg

def information_publisher():

    rospy.init_node('information_publisher', anonymous=True)# ROS node
initialization

    info_pub = rospy.Publisher('/company_info', Information, queue_size=6)

    rate = rospy.Rate(10) #Set the frequency of the loop

    while not rospy.is_shutdown():

    # Initialize learning_topic:: Information type message
        info_msg = Information()
        info_msg.company = "Yahboom";
        info_msg.city  = "Shenzhen";

        info_pub.publish(info_msg)# Publish Message

        rospy.loginfo("This is %s in %s.", info_msg.company, info_msg.city)#
Print and publish messages

        rate.sleep()# Delay according to the cycle frequency

if __name__ == '__main__':
    try:
        information_publisher()
    except rospy.ROSInterruptException:
        pass
```

Information_subscriber.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
```

```
import rospy

from learning_topic.msg import Information #Import custom msg

def CompanyInfoCallback(msg):

    rospy.loginfo("company: name:%s  city:%s ", msg.company, msg.city)#Print
subscription received information

def Infomation_subscriber():

    rospy.init_node('Infomation_subscriber', anonymous=True)# ROS node
initialization

    # Create a subscriber to subscribe to a topic named/company info and register
the callback function personInfocallbacks
    rospy.Subscriber("/company_info", Information, CompanyInfoCallback)

    rospy.spin()# Loop waiting callback function

if __name__ == '__main__':
    Infomation_subscriber()
```

2. Core part

Here is mainly an explanation of how to import custom message modules and how to use them:Import

```
from learning_topic.msg import Information
```

use

```
info_msg = Person()
info_msg.company = "Yahboom";
info_msg.city  = "Shenzhen";
```

3. run a program

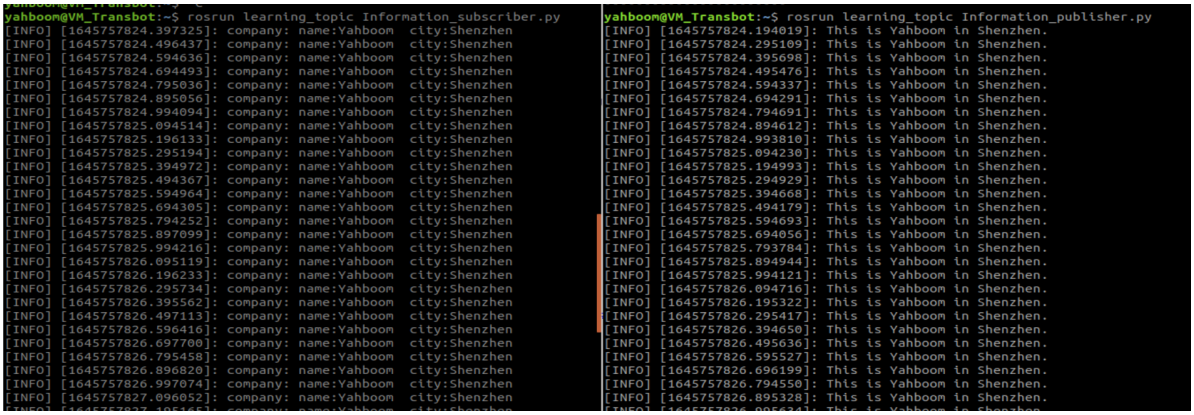
Before running the program, add executable permissions to the py file

```
sudo chmod a+x Information_subscriber.py
sudo chmod a+x Information_publisher.py
```

run a program

```
roscore
rosrun learning_topic Information_publisher.py
rosrun learning_topic Information_subscriber.py
```

4. Run screenshot


yahboom@VM_Transbot:~$ rosrun learning_topic Information_subscriber.py
[INFO] [1645757824.397325]: company: name:Yahboom city:Shenzhen
[INFO] [1645757824.496437]: company: name:Yahboom city:Shenzhen
[INFO] [1645757824.594636]: company: name:Yahboom city:Shenzhen
[INFO] [1645757824.694493]: company: name:Yahboom city:Shenzhen
[INFO] [1645757824.795036]: company: name:Yahboom city:Shenzhen
[INFO] [1645757824.895056]: company: name:Yahboom city:Shenzhen
[INFO] [1645757824.994094]: company: name:Yahboom city:Shenzhen
[INFO] [1645757825.094514]: company: name:Yahboom city:Shenzhen
[INFO] [1645757825.196133]: company: name:Yahboom city:Shenzhen
[INFO] [1645757825.295194]: company: name:Yahboom city:Shenzhen
[INFO] [1645757825.394972]: company: name:Yahboom city:Shenzhen
[INFO] [1645757825.494367]: company: name:Yahboom city:Shenzhen
[INFO] [1645757825.594964]: company: name:Yahboom city:Shenzhen
[INFO] [1645757825.694305]: company: name:Yahboom city:Shenzhen
[INFO] [1645757825.794252]: company: name:Yahboom city:Shenzhen
[INFO] [1645757825.897099]: company: name:Yahboom city:Shenzhen
[INFO] [1645757825.994216]: company: name:Yahboom city:Shenzhen
[INFO] [1645757826.095119]: company: name:Yahboom city:Shenzhen
[INFO] [1645757826.196233]: company: name:Yahboom city:Shenzhen
[INFO] [1645757826.295734]: company: name:Yahboom city:Shenzhen
[INFO] [1645757826.395562]: company: name:Yahboom city:Shenzhen
[INFO] [1645757826.497113]: company: name:Yahboom city:Shenzhen
[INFO] [1645757826.596416]: company: name:Yahboom city:Shenzhen
[INFO] [1645757826.697700]: company: name:Yahboom city:Shenzhen
[INFO] [1645757826.795458]: company: name:Yahboom city:Shenzhen
[INFO] [1645757826.896820]: company: name:Yahboom city:Shenzhen
[INFO] [1645757826.997074]: company: name:Yahboom city:Shenzhen
[INFO] [1645757827.096052]: company: name:Yahboom city:Shenzhen
yahboom@VM_Transbot:~$ rosrun learning_topic Information_publisher.py
[INFO] [1645757824.194019]: This is Yahboom in Shenzhen.
[INFO] [1645757824.295109]: This is Yahboom in Shenzhen.
[INFO] [1645757824.395698]: This is Yahboom in Shenzhen.
[INFO] [1645757824.495476]: This is Yahboom in Shenzhen.
[INFO] [1645757824.594337]: This is Yahboom in Shenzhen.
[INFO] [1645757824.694291]: This is Yahboom in Shenzhen.
[INFO] [1645757824.794691]: This is Yahboom in Shenzhen.
[INFO] [1645757824.894612]: This is Yahboom in Shenzhen.
[INFO] [1645757824.993810]: This is Yahboom in Shenzhen.
[INFO] [1645757825.094230]: This is Yahboom in Shenzhen.
[INFO] [1645757825.194993]: This is Yahboom in Shenzhen.
[INFO] [1645757825.294929]: This is Yahboom in Shenzhen.
[INFO] [1645757825.394668]: This is Yahboom in Shenzhen.
[INFO] [1645757825.494179]: This is Yahboom in Shenzhen.
[INFO] [1645757825.594693]: This is Yahboom in Shenzhen.
[INFO] [1645757825.694056]: This is Yahboom in Shenzhen.
[INFO] [1645757825.793784]: This is Yahboom in Shenzhen.
[INFO] [1645757825.894944]: This is Yahboom in Shenzhen.
[INFO] [1645757825.994121]: This is Yahboom in Shenzhen.
[INFO] [1645757826.094716]: This is Yahboom in Shenzhen.
[INFO] [1645757826.195322]: This is Yahboom in Shenzhen.
[INFO] [1645757826.295417]: This is Yahboom in Shenzhen.
[INFO] [1645757826.394650]: This is Yahboom in Shenzhen.
[INFO] [1645757826.495636]: This is Yahboom in Shenzhen.
[INFO] [1645757826.595527]: This is Yahboom in Shenzhen.
[INFO] [1645757826.696199]: This is Yahboom in Shenzhen.
[INFO] [1645757826.794550]: This is Yahboom in Shenzhen.
[INFO] [1645757826.895328]: This is Yahboom in Shenzhen.