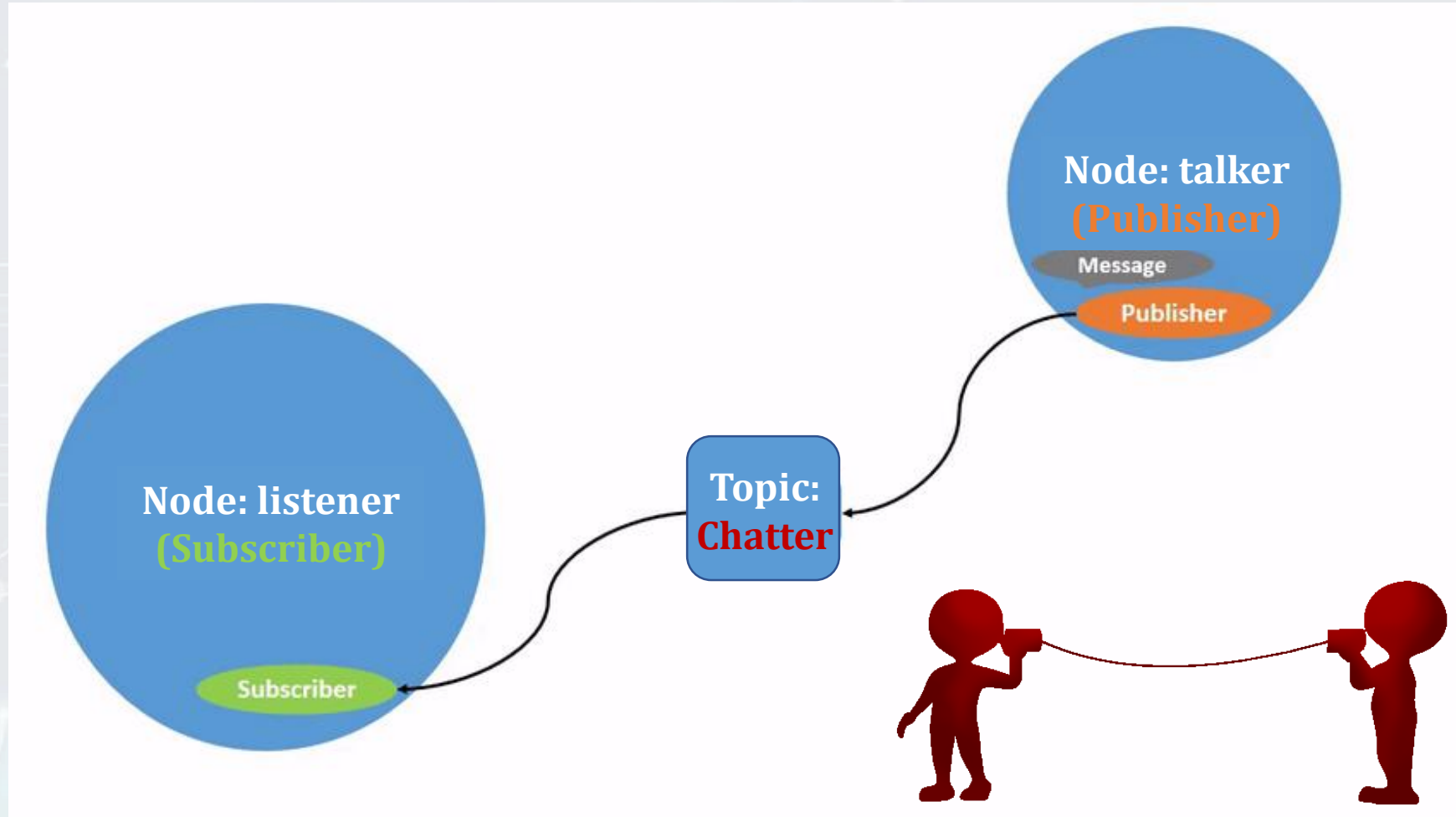


AI ROS 2 Python Chatter By TESR

ROS2

Chatter concept



Create the **workspace** and **package**

- Before we start, Install a useful tool using:

```
sudo pip3 install gdown -y
```

- And then, To create a new ros2 **workspace** open new terminal and type:

```
mkdir -p ros2_ws/src
```

```
rengy@tesr-9939:~$ ls
arduino          catkin_ws        grafana          NodeRed          sketchbook      untitled1
build-test1-Desktop-Debug  Downloads        model_editor_models  python_chatter   turtlebot3
cartographer_ws  foxy_ws          Music            ros2_ws          ui
```

- Change the current working directory to **ros2_ws/src** using:

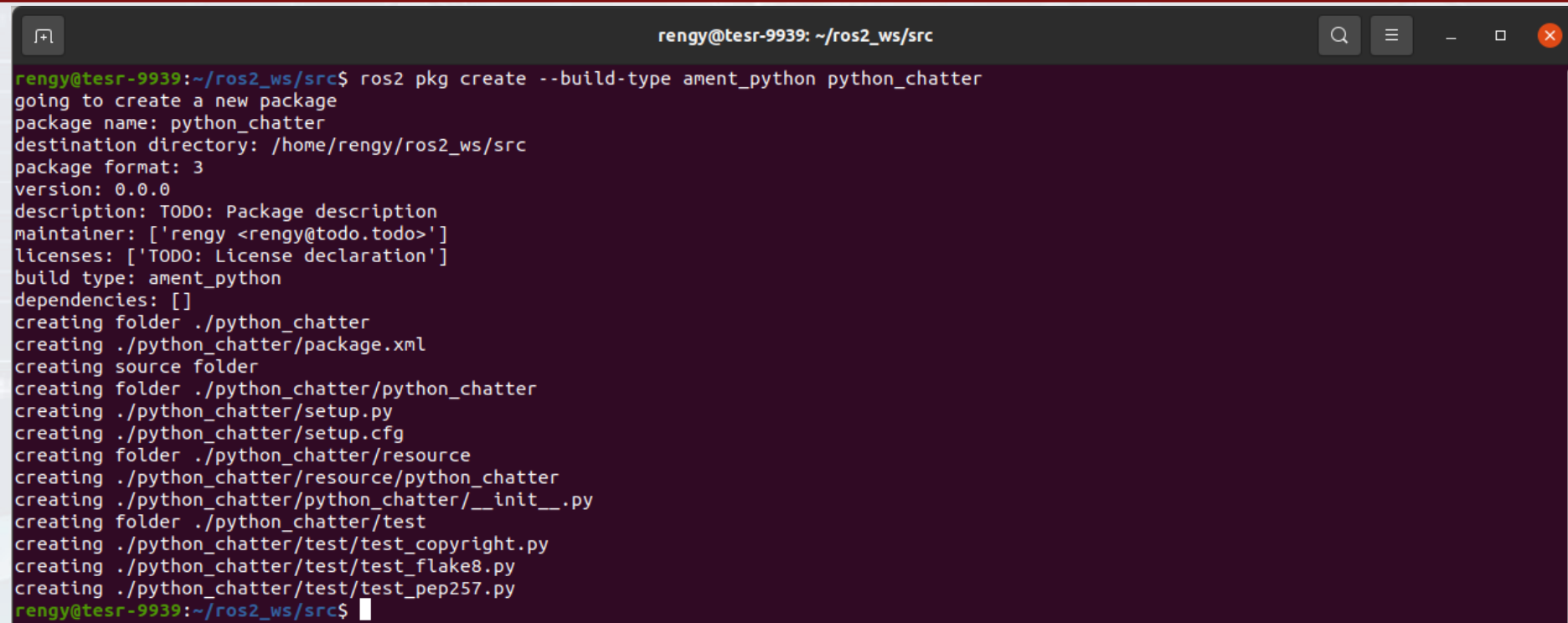
```
cd ros2_ws/src
```

```
rengy@tesr-9939: ~/ros2_ws/src
rengy@tesr-9939:~$ cd ros2_ws/src/
rengy@tesr-9939:~/ros2_ws/src$
```

Create the **workspace** and **package**

- Create the **package** in src of workspace using:

```
ros2 pkg create --build-type ament_python python_chatter
```



```
rengy@tesr-9939: ~/ros2_ws/src
rengy@tesr-9939:~/ros2_ws/src$ ros2 pkg create --build-type ament_python python_chatter
going to create a new package
package name: python_chatter
destination directory: /home/rengy/ros2_ws/src
package format: 3
version: 0.0.0
description: TODO: Package description
maintainer: ['rengy <rengy@todo.todo>']
licenses: ['TODO: License declaration']
build type: ament_python
dependencies: []
creating folder ./python_chatter
creating ./python_chatter/package.xml
creating source folder
creating folder ./python_chatter/python_chatter
creating ./python_chatter/setup.py
creating ./python_chatter/setup.cfg
creating folder ./python_chatter/resource
creating ./python_chatter/resource/python_chatter
creating ./python_chatter/python_chatter/__init__.py
creating folder ./python_chatter/test
creating ./python_chatter/test/test_copyright.py
creating ./python_chatter/test/test_flake8.py
creating ./python_chatter/test/test_pep257.py
rengy@tesr-9939:~/ros2_ws/src$
```

Create the **workspace** and **package**

- And then, you have 2 files that need to edit are:
 - **package.xml**
 - **setup.py**

```
rengy@tesr-9939:~/ros2_ws/src$ ls
python_chatter
rengy@tesr-9939:~/ros2_ws/src$ cd python_chatter/
rengy@tesr-9939:~/ros2_ws/src/python_chatter$ ls
package.xml python_chatter resource setup.cfg setup.py test
```

Create the **workspace** and **package**

- We must add following dependencies corresponding to **package.xml**

`<depend>rclpy</depend>`
`<depend>std_msgs</depend>`



```
1 <?xml version="1.0"?>
2 <?xml-model href="http://download.ros.org/schema/package_format3.xsd"
  schematypens="http://www.w3.org/2001/XMLSchema"?>
3 <package format="3">
4   <name>python_chatter</name>
5   <version>0.0.0</version>
6   <description>TODO: Package description</description>
7   <maintainer email="pi@todo.todo">pi</maintainer>
8   <license>TODO: License declaration</license>
9
10  <buildtool_depend>ament_cmake_python</buildtool_depend>
11  <depend>rclpy</depend>
12  <depend>std_msgs</depend>
13
14  <test_depend>ament_copyright</test_depend>
15  <test_depend>ament_flake8</test_depend>
16  <test_depend>ament_pep257</test_depend>
17  <test_depend>python3-pytest</test_depend>
18
19  <export>
20    <build_type>ament_python</build_type>
21  </export>
22 </package>
```

Create the **workspace** and **package**

- Add following **entry_points** to your node import statement in **setup.py**

```
entry_points={  
    'console_scripts': [  
        'talker = python_chatter.py_talker:main',  
        'listener = python_chatter.py_listener:main',  
    ],  
},
```

```
1 from setuptools import setup  
2  
3 package_name = 'python_chatter'  
4  
5 setup(  
6     name=package_name,  
7     version='0.0.0',  
8     packages=[package_name],  
9     data_files=[  
10         ('share/ament_index/resource_index/packages',  
11          ['resource/' + package_name]),  
12         ('share/' + package_name, ['package.xml']),  
13     ],  
14     install_requires=['setuptools'],  
15     zip_safe=True,  
16     maintainer='pi',  
17     maintainer_email='pi@todo.todo',  
18     description='TODO: Package description',  
19     license='TODO: License declaration',  
20     tests_require=['pytest'],  
21     entry_points={  
22         'console_scripts': [  
23             'talker = python_chatter.py_talker:main',  
24             'listener = python_chatter.py_listener:main',  
25         ],  
26     },  
27 )
```

Build **package**

- Check for missing dependencies before building:

```
cd ~/ros2_ws/  
rosdep install -i --from-path src --rosdistro foxy -y
```

- As a result, show **"All required rosdeps installed successfully"**.

```
rengy@tesr-9939:~/ros2_ws$ rosdep install -i --from-path src --rosdistro foxy -y  
#All required rosdeps installed successfully
```


Build **package**

- So, you are ready to build package using:

```
colcon build
```

```
rengy@tesr-9939:~/ros2_ws$ colcon build
Starting >>> python_chatter
Finished <<< python_chatter [0.82s]

Summary: 1 package finished [0.96s]
```

- After, build is completed source environment and echo source script to

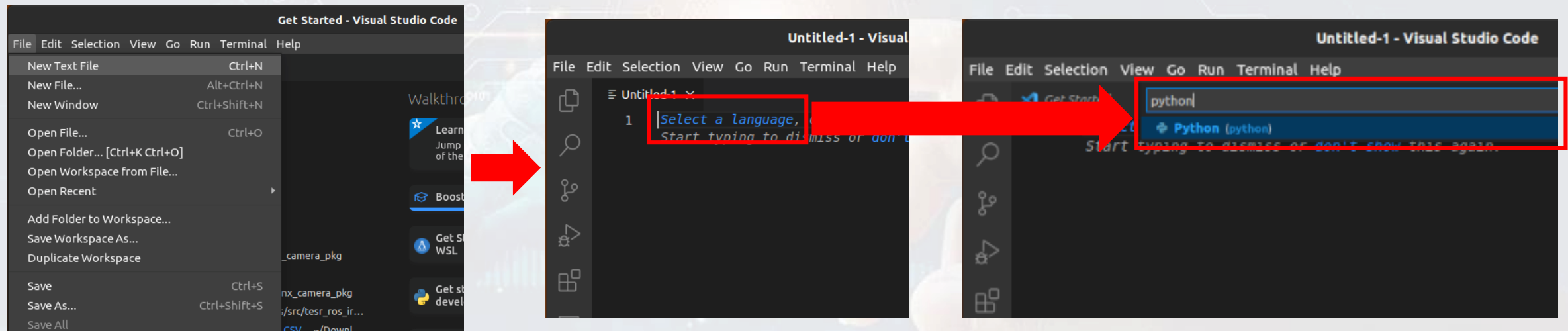
```
echo "source ~/ros2_ws/install/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

Create the Python scripts

- Open the Visual Studio Code using:

code

- Create a new file by select "**File > New Text File**" and select **Python** language.



Create Node Talker(Publisher)

- Example Python code: (name as py_talker.py)

```
#!/usr/bin/python3
import rclpy
from rclpy.node import Node
from std_msgs.msg import String

class talker(Node):
    def __init__(self):
        super().__init__('talker_node')
        self.topic = "chatter"
        self.publishers_ = self.create_publisher(String,self.topic,10)
        timer_peroid = 0.5 #seconds
        self.timer = self.create_timer(timer_peroid, self.timer_callback)
        self.i = 0

    def timer_callback(self):
        msg = String()
        msg.data = "message:" +str(self.i)
        self.publishers_.publish(msg)
        self.get_logger().info('Publishing: "%s"'%msg.data)
        self.i += 1
```

```
def main(args=None):
    rclpy.init(args=args)

    talker_pub = talker()
    rclpy.spin(talker_pub)

    talker_pub.destroy_node()
    rclpy.shutdown()

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

Create Node Listener(Subscriber)

- Example Python code: (name as py_listener.py)

```
#!/usr/bin/python3
import rclpy
from rclpy.node import Node
from std_msgs.msg import String

class listener(Node):
    def __init__(self):
        super().__init__('listener_node')
        self.topic = "chatter"
        self.subscription = self.create_subscription(String, self.topic, self.listener_callback, 10)
        self.subscription

    def listener_callback(self, msg):
        self.get_logger().info('I heard: "%s"' % msg.data)
```

```
def main(args=None):
    rclpy.init(args=args)

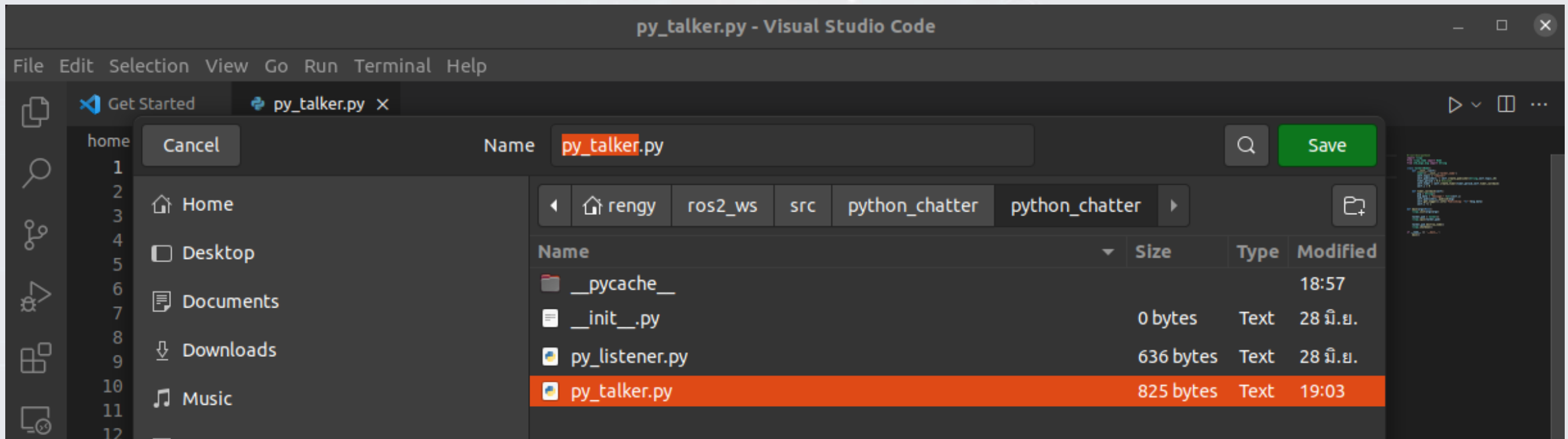
    listener_sub = listener()
    rclpy.spin(listener_sub)

    listener_sub.destroy_node()
    rclpy.shutdown()

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

Save file to **package's** script folder

- Save your code at "**ros2_ws/src/python_chatter/python_chatter**"



Save file to **package's** script folder

- And then, give the permission to execute to the files using:

```
cd ~/ros2_ws/src/python_chatter/python_chatter/  
sudo chmod +x *
```

```
rengy@tesr-9939:~$ cd ~/ros2_ws/src/python_chatter/python_chatter/  
rengy@tesr-9939:~/ros2_ws/src/python_chatter/python_chatter$ sudo chmod +x *  
[sudo] password for rengy:  
rengy@tesr-9939:~/ros2_ws/src/python_chatter/python_chatter$ ls  
__init__.py  __pycache__  py_listener.py  py_talker.py
```

- After that, build the package again using:

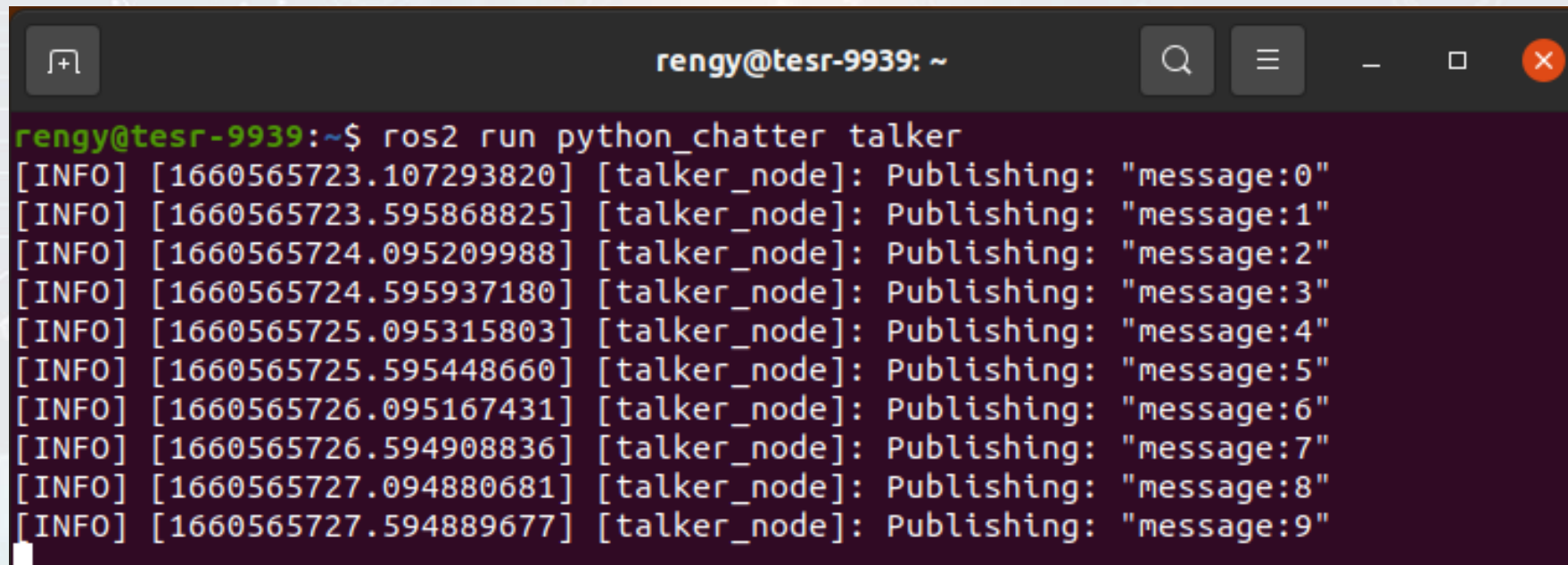
```
cd ~/ros2_ws  
colcon build
```

Run **Talker** using **Python**

- On first terminal, run the talker by type:

```
ros2 run python_chatter talker
```

- As a result, this program is keep publish the "**message: increasing number**"

A terminal window titled 'rengy@tesr-9939: ~' with standard window controls. The terminal shows the command 'ros2 run python_chatter talker' being executed. The output consists of ten lines of log messages, each starting with '[INFO]' followed by a timestamp and the text '[talker_node]: Publishing: "message:0"' through 'message:9'.

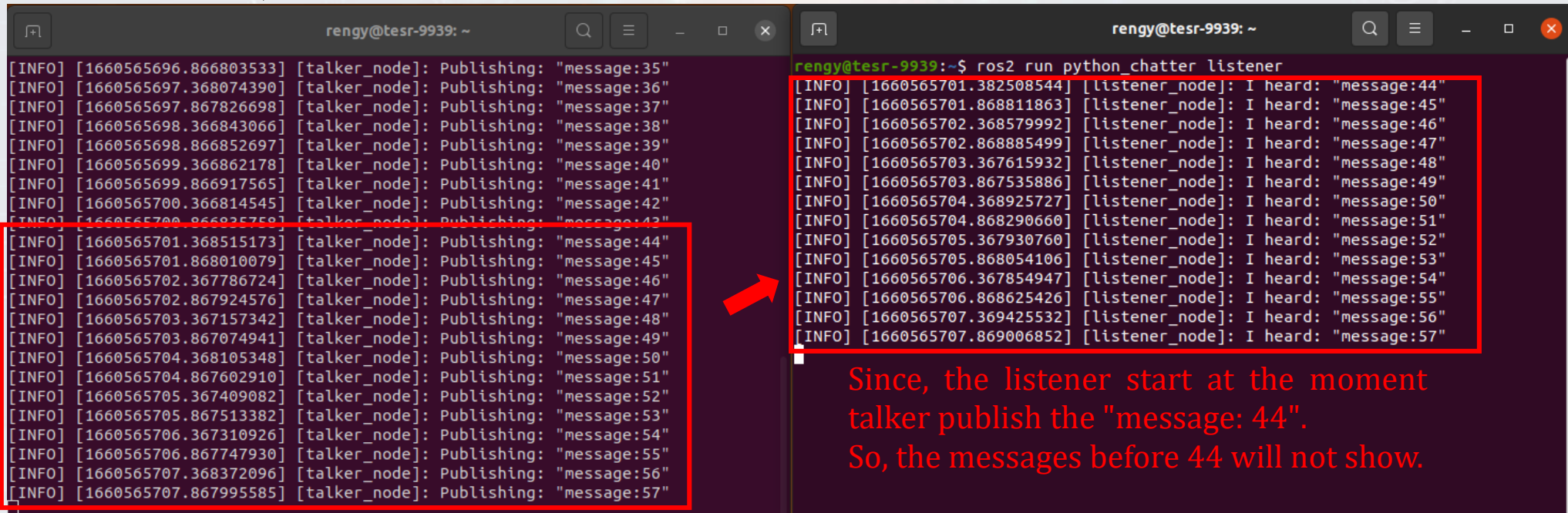
```
rengy@tesr-9939:~$ ros2 run python_chatter talker
[INFO] [1660565723.107293820] [talker_node]: Publishing: "message:0"
[INFO] [1660565723.595868825] [talker_node]: Publishing: "message:1"
[INFO] [1660565724.095209988] [talker_node]: Publishing: "message:2"
[INFO] [1660565724.595937180] [talker_node]: Publishing: "message:3"
[INFO] [1660565725.095315803] [talker_node]: Publishing: "message:4"
[INFO] [1660565725.595448660] [talker_node]: Publishing: "message:5"
[INFO] [1660565726.095167431] [talker_node]: Publishing: "message:6"
[INFO] [1660565726.594908836] [talker_node]: Publishing: "message:7"
[INFO] [1660565727.094880681] [talker_node]: Publishing: "message:8"
[INFO] [1660565727.594889677] [talker_node]: Publishing: "message:9"
```


Run Listener using Python

- Open the second terminal and then run the listener by type:

```
ros2 run python_chatter listener
```

- As a result, the second terminal will show from talker:



```
rengy@tesr-9939: ~  
[INFO] [1660565696.866803533] [talker_node]: Publishing: "message:35"  
[INFO] [1660565697.368074390] [talker_node]: Publishing: "message:36"  
[INFO] [1660565697.867826698] [talker_node]: Publishing: "message:37"  
[INFO] [1660565698.366843066] [talker_node]: Publishing: "message:38"  
[INFO] [1660565698.866852697] [talker_node]: Publishing: "message:39"  
[INFO] [1660565699.366862178] [talker_node]: Publishing: "message:40"  
[INFO] [1660565699.866917565] [talker_node]: Publishing: "message:41"  
[INFO] [1660565700.366814545] [talker_node]: Publishing: "message:42"  
[INFO] [1660565700.866825759] [talker_node]: Publishing: "message:43"  
[INFO] [1660565701.368515173] [talker_node]: Publishing: "message:44"  
[INFO] [1660565701.868010079] [talker_node]: Publishing: "message:45"  
[INFO] [1660565702.367786724] [talker_node]: Publishing: "message:46"  
[INFO] [1660565702.867924576] [talker_node]: Publishing: "message:47"  
[INFO] [1660565703.367157342] [talker_node]: Publishing: "message:48"  
[INFO] [1660565703.867074941] [talker_node]: Publishing: "message:49"  
[INFO] [1660565704.368105348] [talker_node]: Publishing: "message:50"  
[INFO] [1660565704.867602910] [talker_node]: Publishing: "message:51"  
[INFO] [1660565705.367409082] [talker_node]: Publishing: "message:52"  
[INFO] [1660565705.867513382] [talker_node]: Publishing: "message:53"  
[INFO] [1660565706.367310926] [talker_node]: Publishing: "message:54"  
[INFO] [1660565706.867747930] [talker_node]: Publishing: "message:55"  
[INFO] [1660565707.368372096] [talker_node]: Publishing: "message:56"  
[INFO] [1660565707.867995585] [talker_node]: Publishing: "message:57"  
  
rengy@tesr-9939: ~$ ros2 run python_chatter listener  
[INFO] [1660565701.382508544] [listener_node]: I heard: "message:44"  
[INFO] [1660565701.868811863] [listener_node]: I heard: "message:45"  
[INFO] [1660565702.368579992] [listener_node]: I heard: "message:46"  
[INFO] [1660565702.868885499] [listener_node]: I heard: "message:47"  
[INFO] [1660565703.367615932] [listener_node]: I heard: "message:48"  
[INFO] [1660565703.867535886] [listener_node]: I heard: "message:49"  
[INFO] [1660565704.368925727] [listener_node]: I heard: "message:50"  
[INFO] [1660565704.868290660] [listener_node]: I heard: "message:51"  
[INFO] [1660565705.367930760] [listener_node]: I heard: "message:52"  
[INFO] [1660565705.868054106] [listener_node]: I heard: "message:53"  
[INFO] [1660565706.367854947] [listener_node]: I heard: "message:54"  
[INFO] [1660565706.868625426] [listener_node]: I heard: "message:55"  
[INFO] [1660565707.369425532] [listener_node]: I heard: "message:56"  
[INFO] [1660565707.869006852] [listener_node]: I heard: "message:57"
```

Since, the listener start at the moment talker publish the "message: 44".
So, the messages before 44 will not show.

ROS 2 element in Chatter

- Check Chatter's ROS 2 node

ros2 node list

- Check Chatter's ROS 2 topic

ros2 topic list

- Check Chatter's ROS 2 service

ros2 service list

- Check Chatter's ROS 2 param

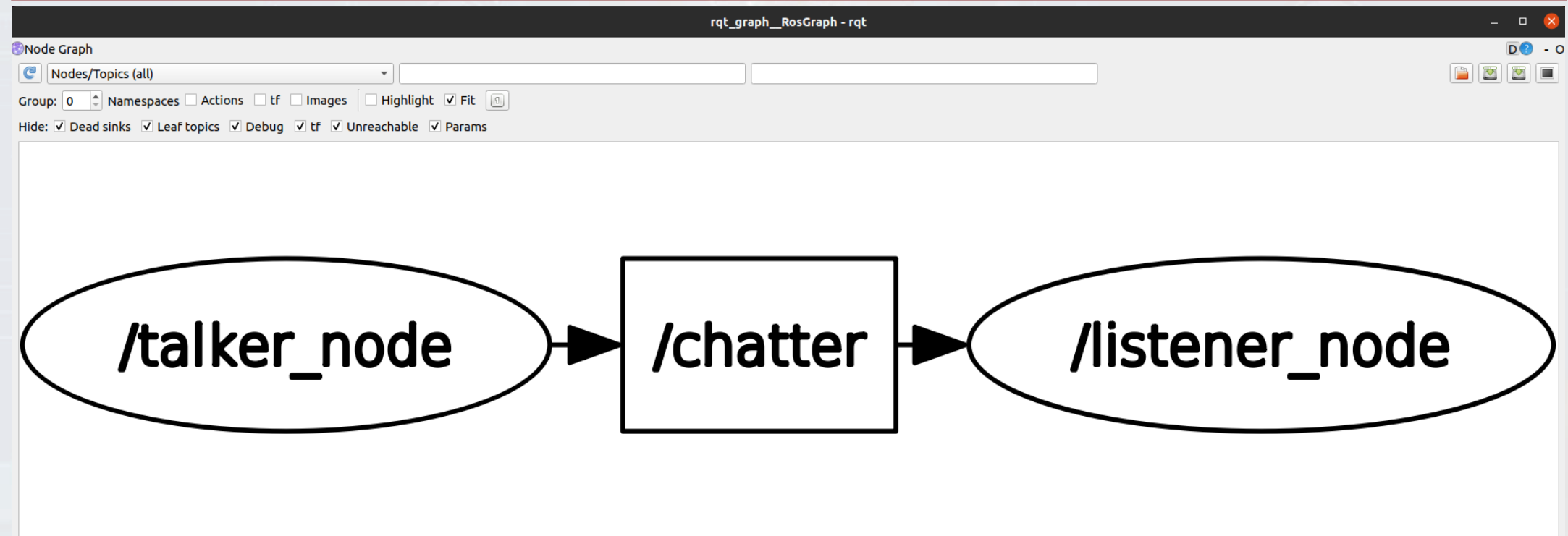
ros2 param list

```
rengy@tesr-9939:~$ ros2 node list
/listener_node
/talker_node
rengy@tesr-9939:~$ ros2 topic list
/chatter
/parameter_events
/rosout
rengy@tesr-9939:~$ ros2 service list
/listener_node/describe_parameters
/listener_node/get_parameter_types
/listener_node/get_parameters
/listener_node/list_parameters
/listener_node/set_parameters
/listener_node/set_parameters_atomically
/talker_node/describe_parameters
/talker_node/get_parameter_types
/talker_node/get_parameters
/talker_node/list_parameters
/talker_node/set_parameters
/talker_node/set_parameters_atomically
rengy@tesr-9939:~$ ros2 param list
/listener_node:
  use_sim_time
/talker_node:
  use_sim_time
```

Chatter's RosGraph by rqt_graph

- Open new terminal and type:

```
rqt_graph
```



Contact Us

Email: tesrshop@gmail.com

Line official Account: @ion1900z

Facebook fanpage: TESR

Tel. 082-983-7768

Scan here



TESR Co., LTD

112/296 หมู่บ้าน เพอร์เฟค มาสเตอร์พีช
หมู่ที่ 2 ตำบลไทรมา อำเภอมะนังนบุรี
จังหวัดนบุรี 11000