

# 無人載具技術與應用

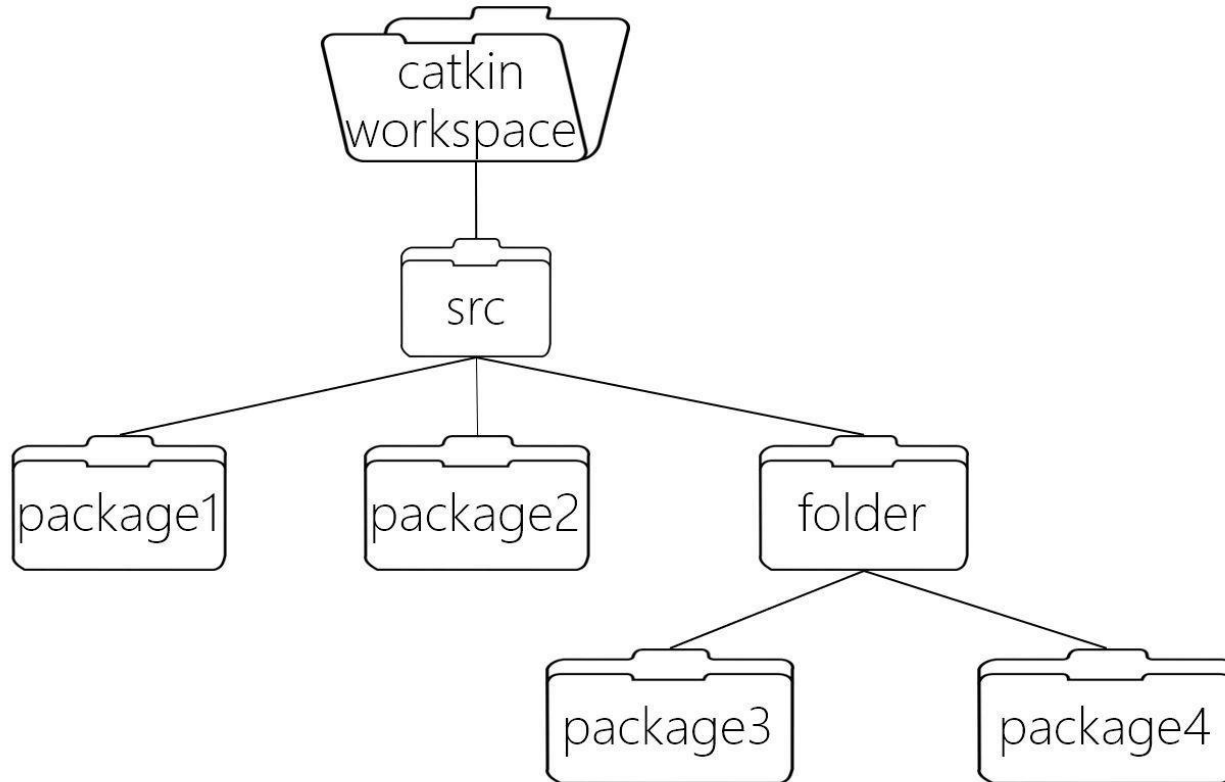
## ROS-Ubuntu

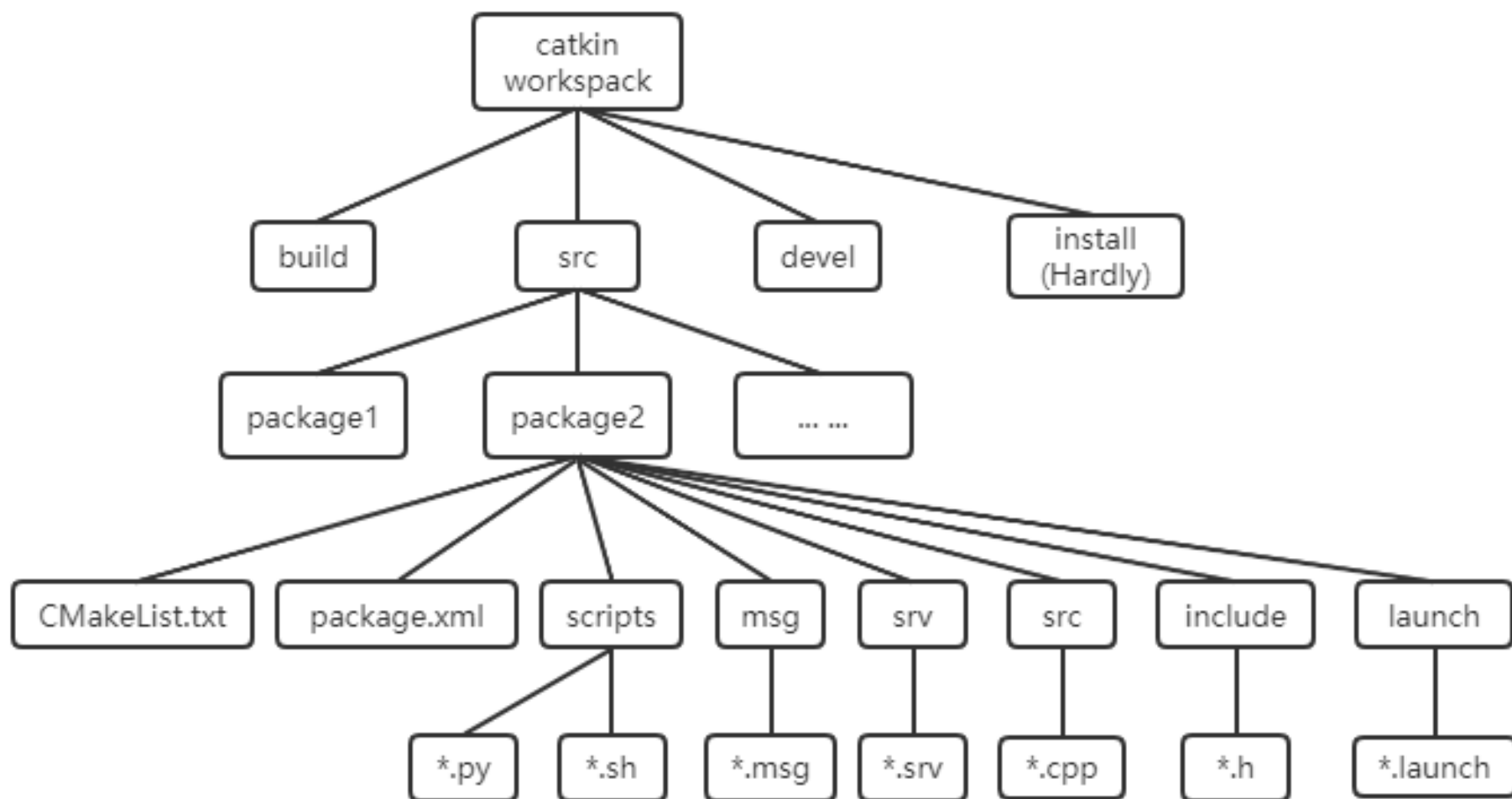
徐瑋隆

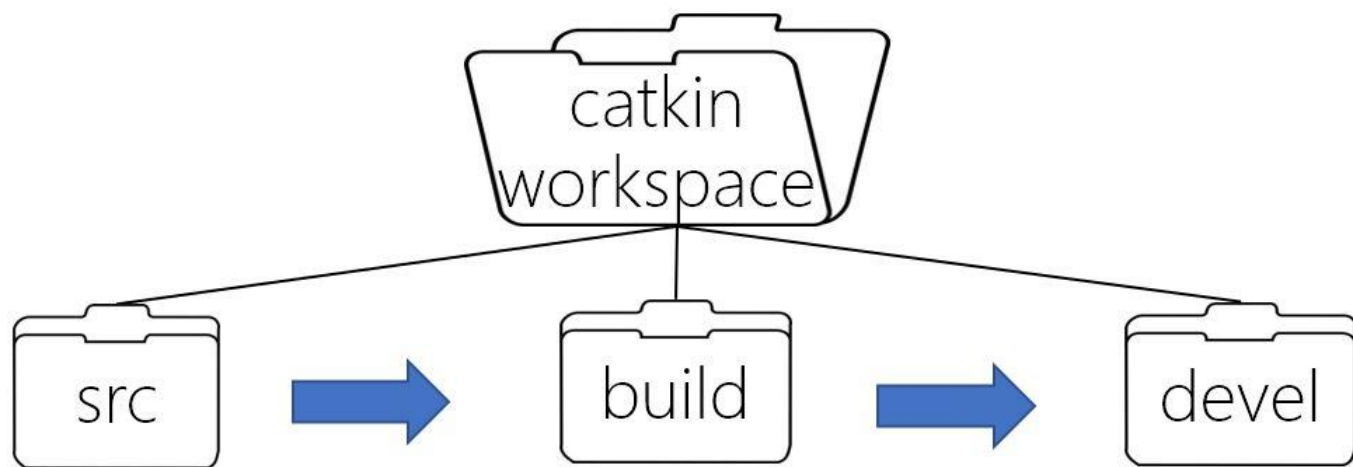
wlhsu304@gmail.com

# 創建 ROS PACKAGE

catkin\_create\_pkg







package源代码包

cmake&catkin缓存和中间文件

目标文件

# 創建 ROS package (catkin\_create\_pkg)

- 工作空間建立(上週已經完成)
- `mkdir -p ~/catkin_ws/src`
- `cd catkin_ws/`
- `catkin_make`
  
- `cd ~/catkin_ws/src`
- `catkin_create_pkg my_work01 std_msgs rospy roscpp`

```
cd ~/catkin_ws/src
```

```
catkin_create_pkg my_work01 std_msgs rospy roscpp
```

```
x - □ peter@peter-lenovo-g50-80: ~  
peter@peter-lenovo-g50-80:~$ cd ~/catkin_ws/src/
```

```
x - □ peter@peter-lenovo-g50-80: ~/catkin_ws/src  
peter@peter-lenovo-g50-80:~/catkin_ws/src$ catkin_create_pkg my_work01 std_msgs  
rospy roscpp
```

```
Created file my_work01/package.xml  
Created file my_work01/CMakeLists.txt  
Created folder my_work01/include/my_work01  
Created folder my_work01/src  
Successfully created files in /home/peter/catkin_ws/src/my_work01. Please adjust  
the values in package.xml.  
peter@peter-lenovo-g50-80:~/catkin_ws/src$
```

每次專案有改變時 皆須執行以下(只改設定檔 則不須)

```
cd ~/catkin_ws
```

```
catkin_make
```

```
source ~/catkin_ws/devel/setup.bash
```

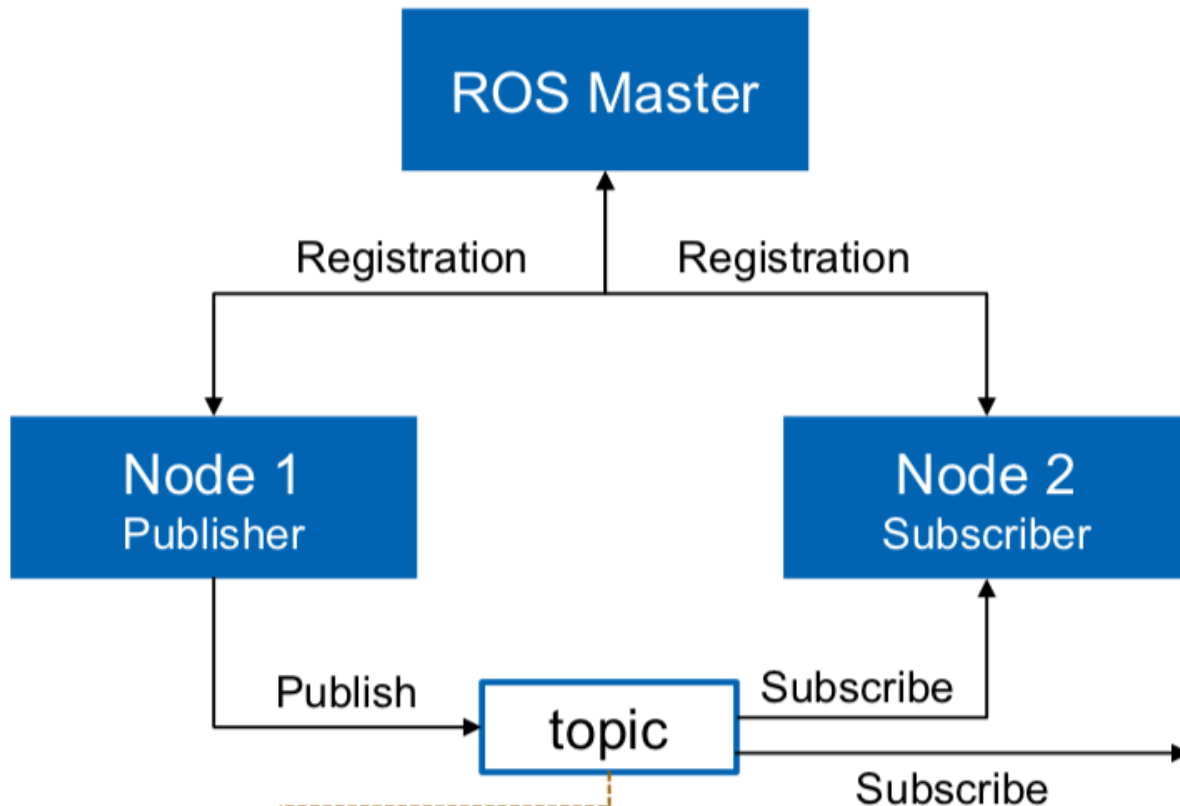
```
peter@peter-lenovo-g50-80:~/catkin_ws/src$ cd ~/catkin_ws/  
peter@peter-lenovo-g50-80:~/catkin_ws$ catkin_make
```

```
peter@peter-lenovo-g50-80:~/catkin_ws$ source ~/catkin_ws/devel/setup.bash
```



**ROS 程式撰寫**

# Publisher與Subscriber talk



\*.msg

Message definition

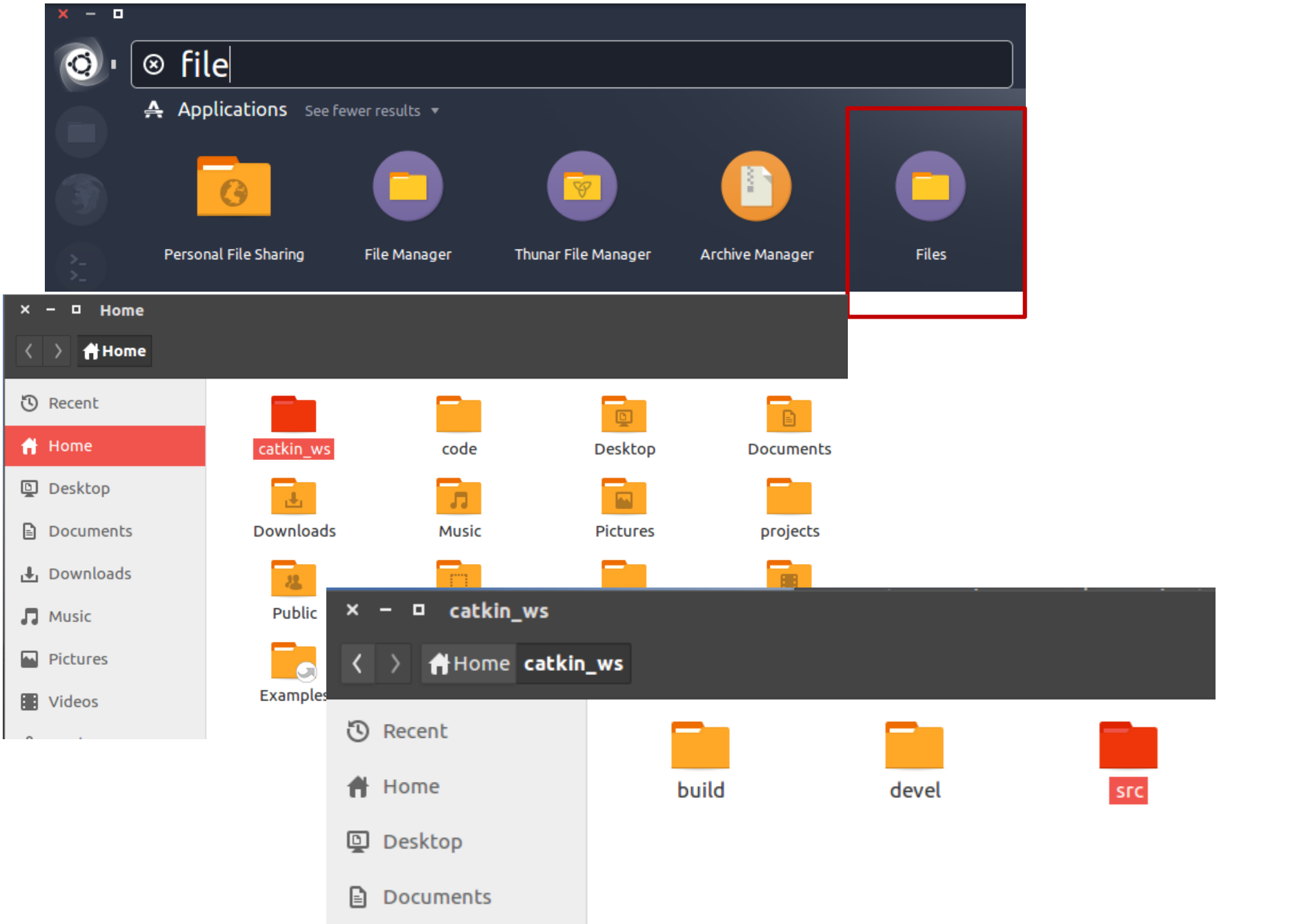
```
int number
double width
string description
etc.
```

**More info**

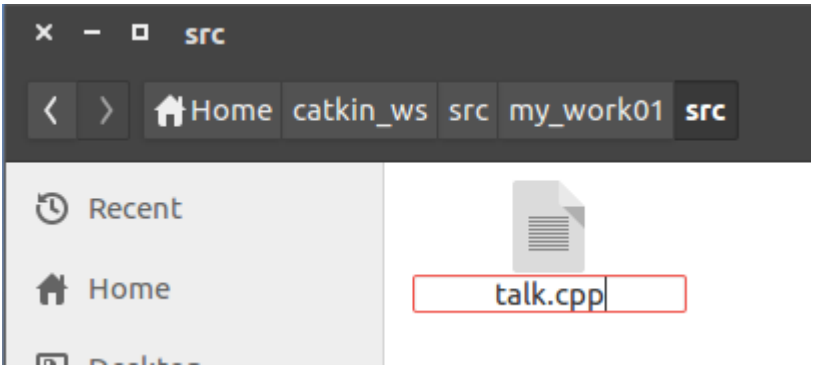
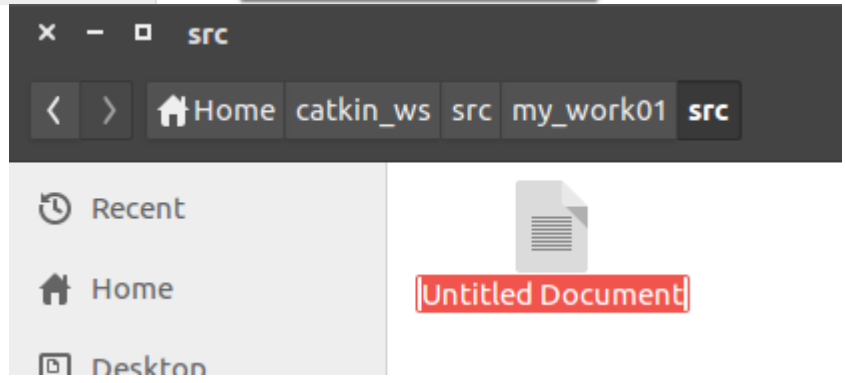
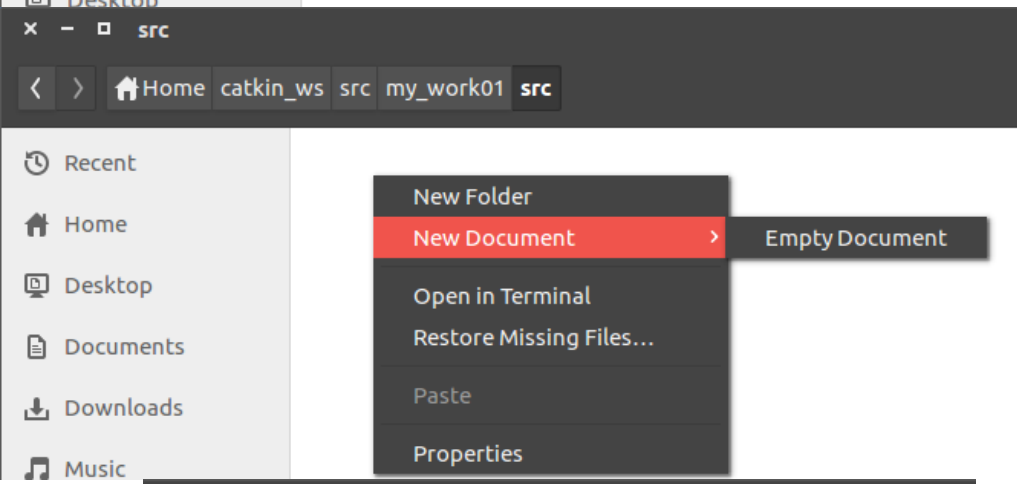
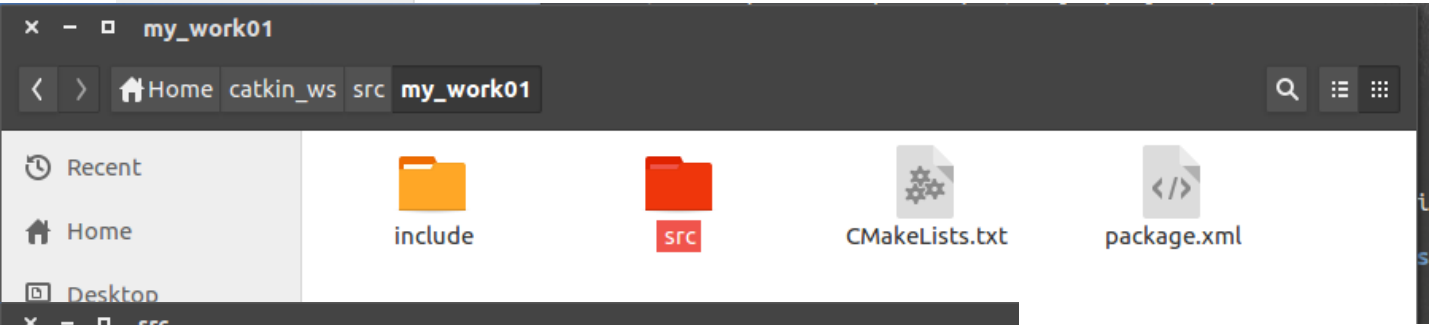
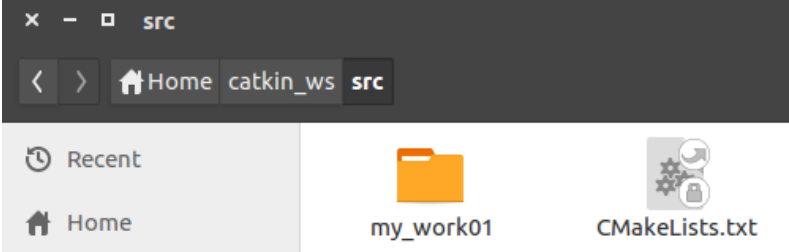
<http://wiki.ros.org/Messages>

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc, char **argv)
{
    ros::init(argc, argv, "talker");
    ros::NodeHandle n;
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
    ros::Rate loop_rate(10); // 10Hz
    int count = 0;
    while (ros::ok())
    {
        std_msgs::String msg;
        std::stringstream ss;
        ss << "hello world " << count;
        msg.data = ss.str();
        ROS_INFO("%s", msg.data.c_str());
        chatter_pub.publish(msg);
        ros::spinOnce();
        loop_rate.sleep();
        ++count;
    }
    return 0;
}
```

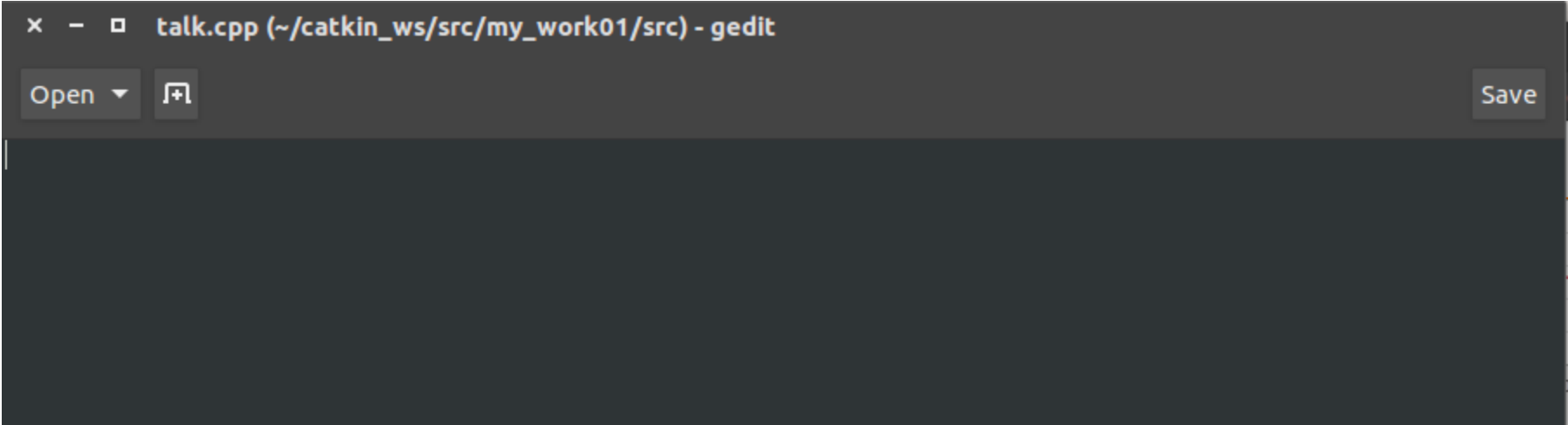
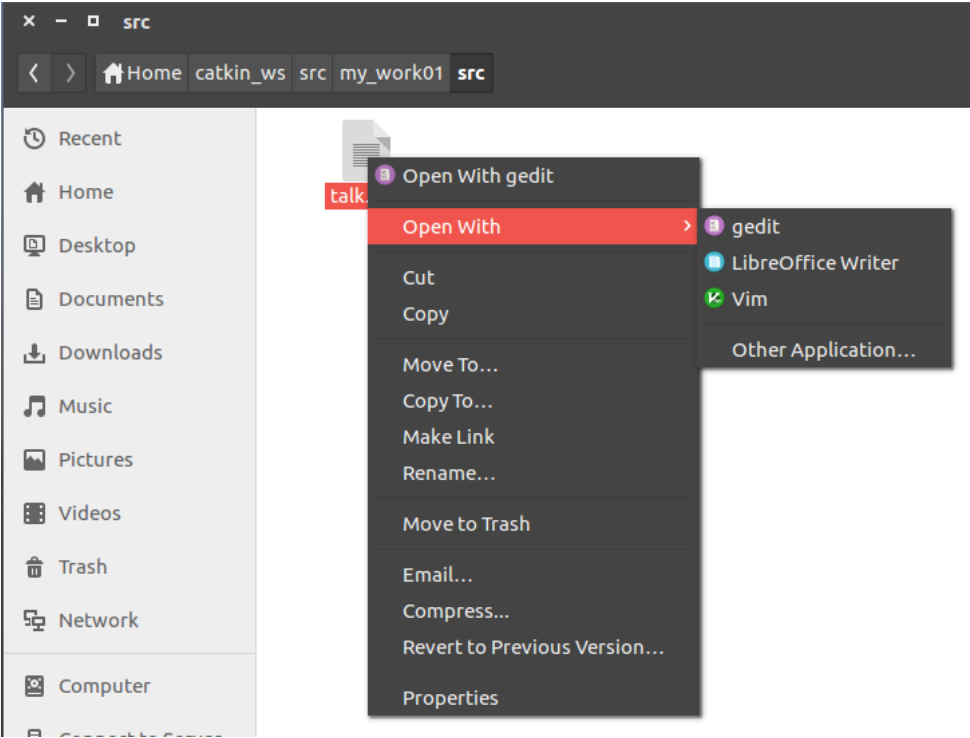
talk



talk



talk



talk

https://github.com/Waywrong/ros\_course

Waywrong/ros\_course

github.com/Waywrong/ros\_course

Search or jump to...

Pull requests

Issues

Marketplace

Explore

Waywrong / ros\_course

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

main

1 branch

0 tags

Go to file

Add file

Code

Waywrong first commit

my\_work01

first commit

Waywrong first commit

...

src

CMakeLists.txt

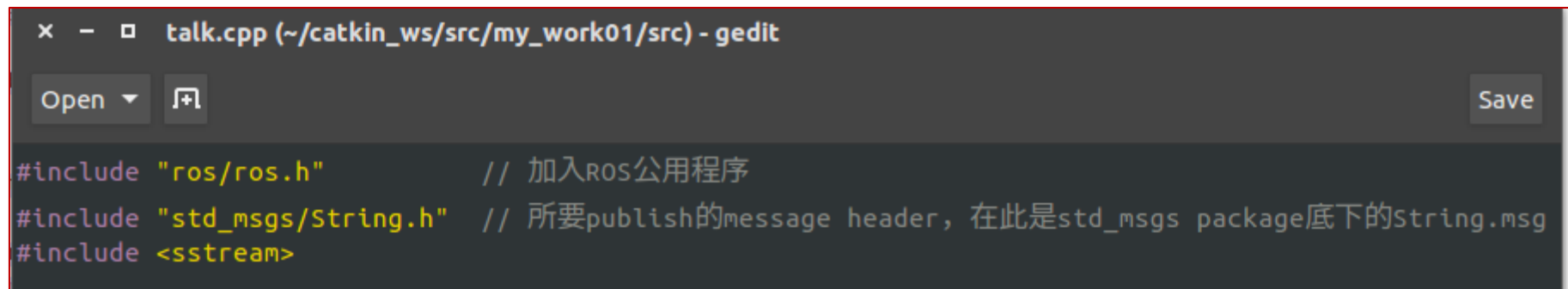
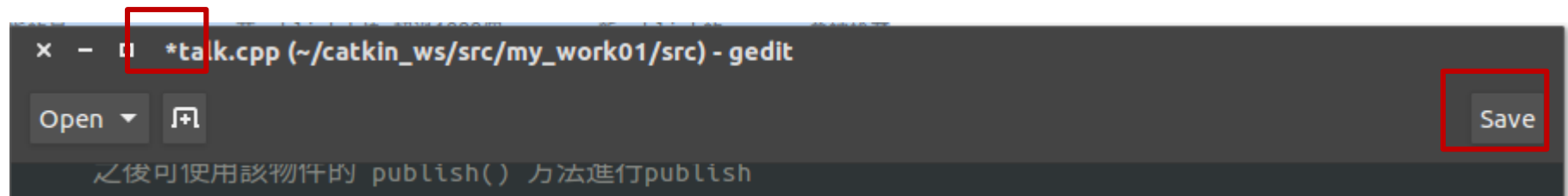
package.xml

talk.cpp

45 lines (33 sloc) | 1.62 KB

```
1 #include "ros/ros.h" // 加入ROS公用程序
2 #include "std_msgs/String.h" // 所要publish的message header, 在此是std_msgs package底下的String.msg
3 #include <sstream>
4
5 int main(int argc, char **argv)
6 {
7     ros::init(argc, argv, "talker"); //一開始必須先初始化, 指定node名稱為talker
8
9     /* 該node與ROS系統通訊的存取點(handle), 建構子會初始化該node,
10      當離開此scope, 解構子會關閉該node */
11     ros::NodeHandle n;
12
13     /* advertise()會將建立topic的資訊告訴master node, 回傳一個Publisher物件(在此為chatter_pub)
14      之後可使用該物件的 publish() 方法進行publish
15      而預計publish的message為std_msgs package的String.msg (std::msgs::String)
16      指定的topic名稱chatter
17      1000指的是message queue, 若publish太快, 超過1000個message, 新publish的message會被排棄
18      */
19     ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);
20
21     ros::Rate loop_rate(10); // 10Hz
22
23     int count = 0;
24     while (ros::ok())
25     {
26         std_msgs::String msg; // 建立暫存的message, 先將資料存入此變數, 再進行publish
27
28         std::stringstream ss;
29         ss << "hello world " << count;
30         msg.data = ss.str(); // 寫入msg message中的data欄位
31
32         ROS_INFO("%s", msg.data.c_str());
33     }
```

talk





## listener

```
#include "ros/ros.h"      // 加入ROS公用程序
#include "std_msgs/String.h" // 所要subscribe的message header，在此是std_msgs
                             package底下的String.msg
```

```
void chatterCallback(const std_msgs::String::ConstPtr& msg)
{
    ROS_INFO("I heard: [%s]", msg->data.c_str());
}
```

```
int main(int argc, char **argv)
{
    ros::init(argc, argv, "listener");
    ros::NodeHandle n;

    ros::Subscriber sub = n.subscribe("chatter", 1000, chatterCallback);

    ros::spin();

    return 0;
}
```

Waywrong / ros\_course

<> Code   ! Issues   🔗 Pull requests   ▶ Actions   📁 Projects   🛡 Security   📊 Insights

main ▾

ros\_course / my\_work01 / **src / talk.cpp**

main ▾

ros\_course / my\_work01 / **src /**

Waywrong listener

..

**listener.cpp**

talk.cpp

```
7  */
8  void chatterCallback(const std_msgs::String& msg)
9  {
10     ROS_INFO("I heard: %s", msg.data.c_str());
11 }
12
13 int main(int argc, char** argv)
14 {
15     ros::init(argc, argv, "chatter");
16
17     /* 該node與ROS主節點建立連接
18     當離開此scope, 解構子會關閉該node */
19     ros::NodeHandle n;
```

Copy

Select All

Search Google for "#include "ros/r..."

View Selection Source

Inspect Element (Q)

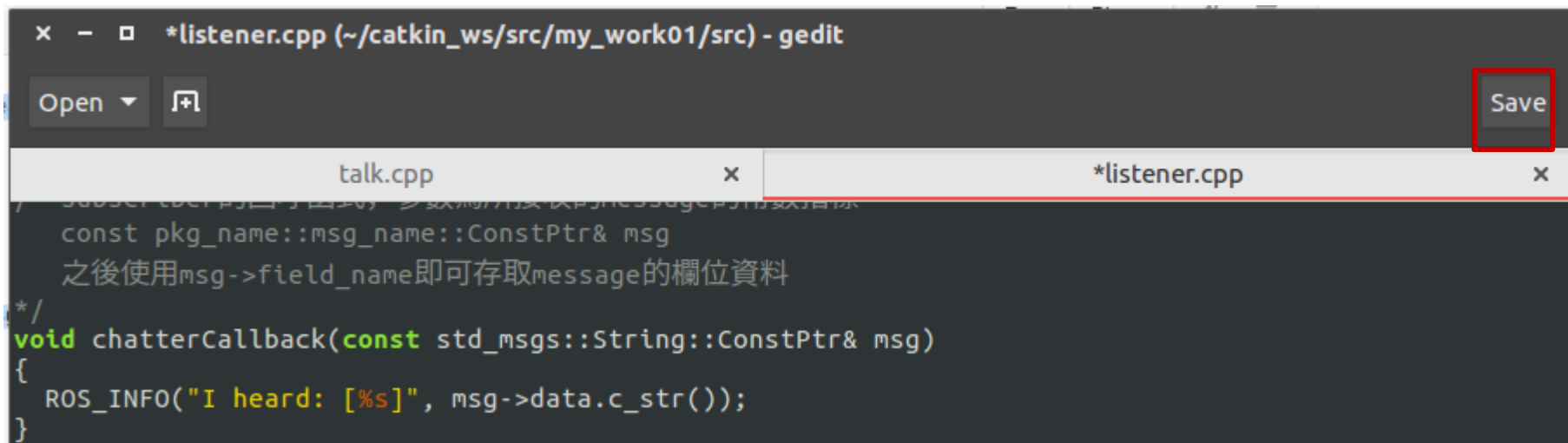
Take a Screenshot

定node名稱為listener

# listener

```
x - □ peter@peter-VirtualBox: ~/catkin_ws  
peter@peter-VirtualBox:~/catkin_ws$ cd ~/catkin_ws/src/my_work01/src/
```

```
x - □ peter@peter-VirtualBox: ~/catkin_ws/src/my_work01/src  
peter@peter-VirtualBox:~/catkin_ws$ cd ~/catkin_ws/src/my_work01/src/  
peter@peter-VirtualBox:~/catkin_ws/src/my_work01/src$ gedit listener.cpp
```



```
x - □ *listener.cpp (~/catkin_ws/src/my_work01/src) - gedit  
Open ▾ [icon] Save  
talk.cpp x *listener.cpp x  
/*  
  subscriber 的回调函数，参数为所接收的message的指针指标  
  之後使用msg->field_name即可存取message的欄位資料  
*/  
void chatterCallback(const std_msgs::String::ConstPtr& msg)  
{  
  ROS_INFO("I heard: [%s]", msg->data.c_str());  
}
```

# CmakeLists.txt

C

```
x - □ peter@peter-VirtualBox: ~  
peter@peter-VirtualBox:~$ cd ~/catkin_ws/src/my_work01/
```

```
x - □ peter@peter-VirtualBox: ~/catkin_ws/src/my_work01  
peter@peter-VirtualBox:~$ cd ~/catkin_ws/src/my_work01/  
peter@peter-VirtualBox:~/catkin_ws/src/my_work01$ gedit CMakeLists.txt
```

```
#####  
## Build ##  
#####  
  
## Specify additional locations of header files  
## Your package locations should be listed before other locations  
include_directories(  
# include  
  ${catkin_INCLUDE_DIRS}  
)  
  
add_executable(talk src/talk.cpp)  
target_link_libraries(talk ${catkin_LIBRARIES})  
add_executable(listener src/listener.cpp)  
target_link_libraries(listener ${catkin_LIBRARIES})
```

每次專案有改變時 皆須執行以下(只改設定檔 則不須)

```
cd ~/catkin_ws
```

```
catkin_make
```

```
source ~/catkin_ws/devel/setup.bash
```

```
peter@peter-lenovo-g50-80:~/catkin_ws/src$ cd ~/catkin_ws/  
peter@peter-lenovo-g50-80:~/catkin_ws$ catkin_make
```

```
peter@peter-lenovo-g50-80:~/catkin_ws$ source ~/catkin_ws/devel/setup.bash
```

# roslaunch

- roscore
- roslaunch my\_work01 talk
- roslaunch my\_work01 listener
- 檢查工具
  - rostopic list
  - rqt\_graph

```
peter@peter-VirtualBox: ~/catkin_ws
peter@peter-Vir... x peter@peter-Vir... x peter@pete
peter@peter-VirtualBox:~/catkin_ws$ roscore
```

source ~/catkin\_ws/devel/setup.bash

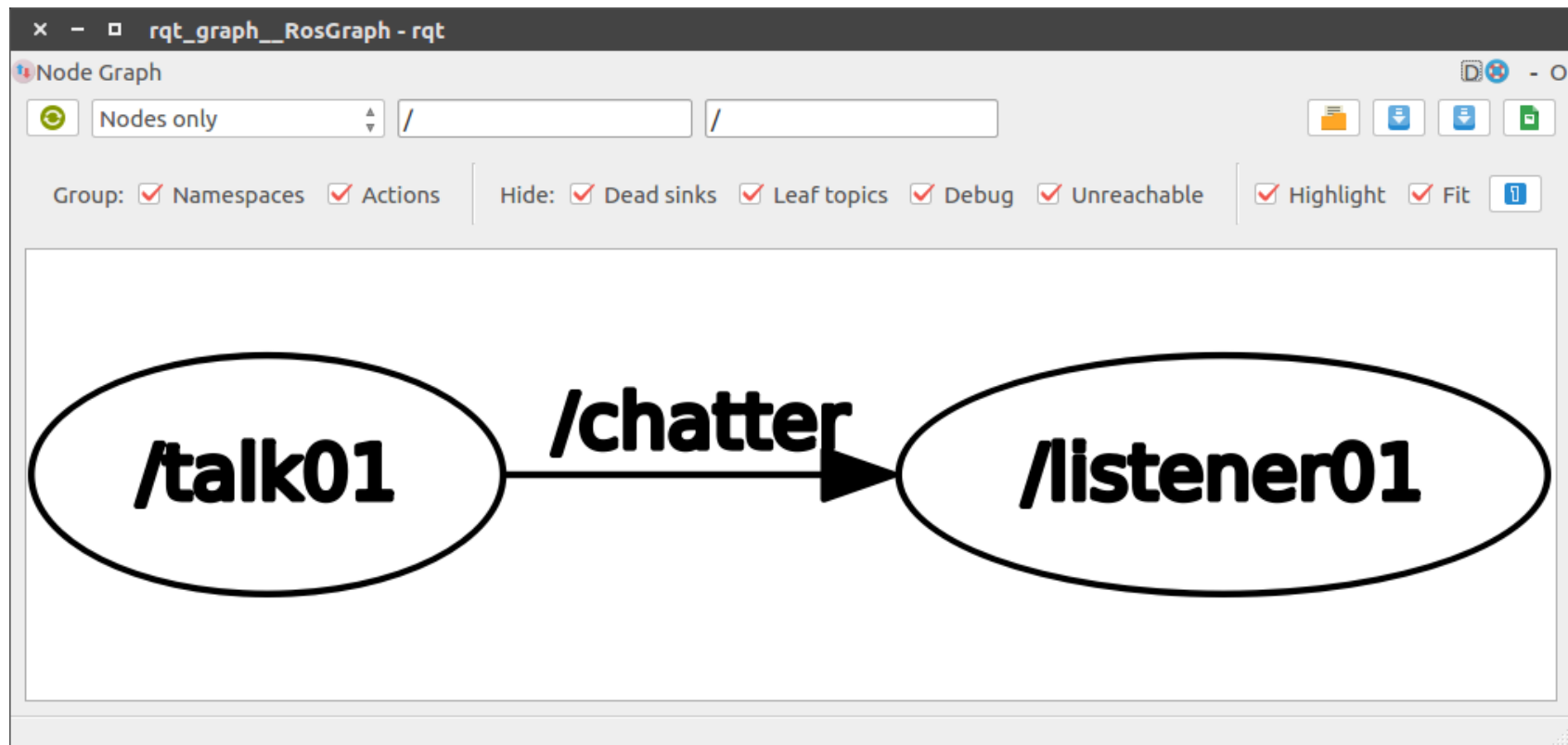
```
peter@peter-VirtualBox: ~/catkin_ws
peter@peter-Vir... x peter@peter-Vir... x peter@peter-Vir... x peter
peter@peter-VirtualBox:~/catkin_ws$ rosrun my_work01 talk
```

```
peter@peter-VirtualBox: ~/catkin_ws
peter@peter-Vir... x peter@peter-Vir... x peter@peter-Vir... x peter@pet
peter@peter-VirtualBox:~/catkin_ws$ rosrun my_work01 listener
```

```
peter@peter-VirtualBox:~/catkin_ws/src/my_work01/launch$ rosnodetool list
/listener01
/rostop
/talk01
```

```
<?xml version="1.0"?>
<launch>
  <node pkg="my_work01"
    name="talk01"
    type="talk01" />
  <node pkg="my_work01"
    name="listener01"
    type="listener01" />
</launch>
```

rqt\_graph





**ROSLAUNCH**

# roslaunch

```
peter@peter-VirtualBox:~$ cd ~/catkin_ws/src/my_work01/
```

```
peter@peter-VirtualBox:~/catkin_ws/src/my_work01$ mkdir launch
```

```
peter@peter-VirtualBox:~/catkin_ws/src/my_work01/launch$ gedit talk.launch
```

```
<?xml version="1.0"?>
<launch>
  <node pkg="my_work01" type="talk" name="talk01" output="screen"/>
</launch>
```

```
peter@peter-VirtualBox:~/catkin_ws/src/my_work01/launch$ roslaunch my_work01 talk.launch
```

## roslaunch 執行多個程式

先停掉前面執行的程式

Ctrl + c

或

rostopic kill talk01

```
peter@peter-VirtualBox:~/catkin_ws/src/my_work01/launch$ rostopic list
/rosout
/talk01
peter@peter-VirtualBox:~/catkin_ws/src/my_work01/launch$ rostopic kill talk01
killing /talk01
killed
```

最後大招 砍roscore

killall -9 rosmaster

```
peter@peter-VirtualBox:~/catkin_ws/src/my_work01/launch$ gedit talk.launch
```

```
<?xml version="1.0"?>
<launch>
  <node pkg="my_work01" type="talk" name="talk01" output="screen"/>
  <node pkg="my_work01" type="listener" name="listener01" output="screen"/>
</launch>
```

```
peter@peter-VirtualBox:~/catkin_ws/src/my_work01/launch$ roslaunch my_work01 talk.launch
```

```
peter@peter-VirtualBox:~/catkin_ws/src/my_work01/launch$ roslaunch my_work01 talk.launch
```

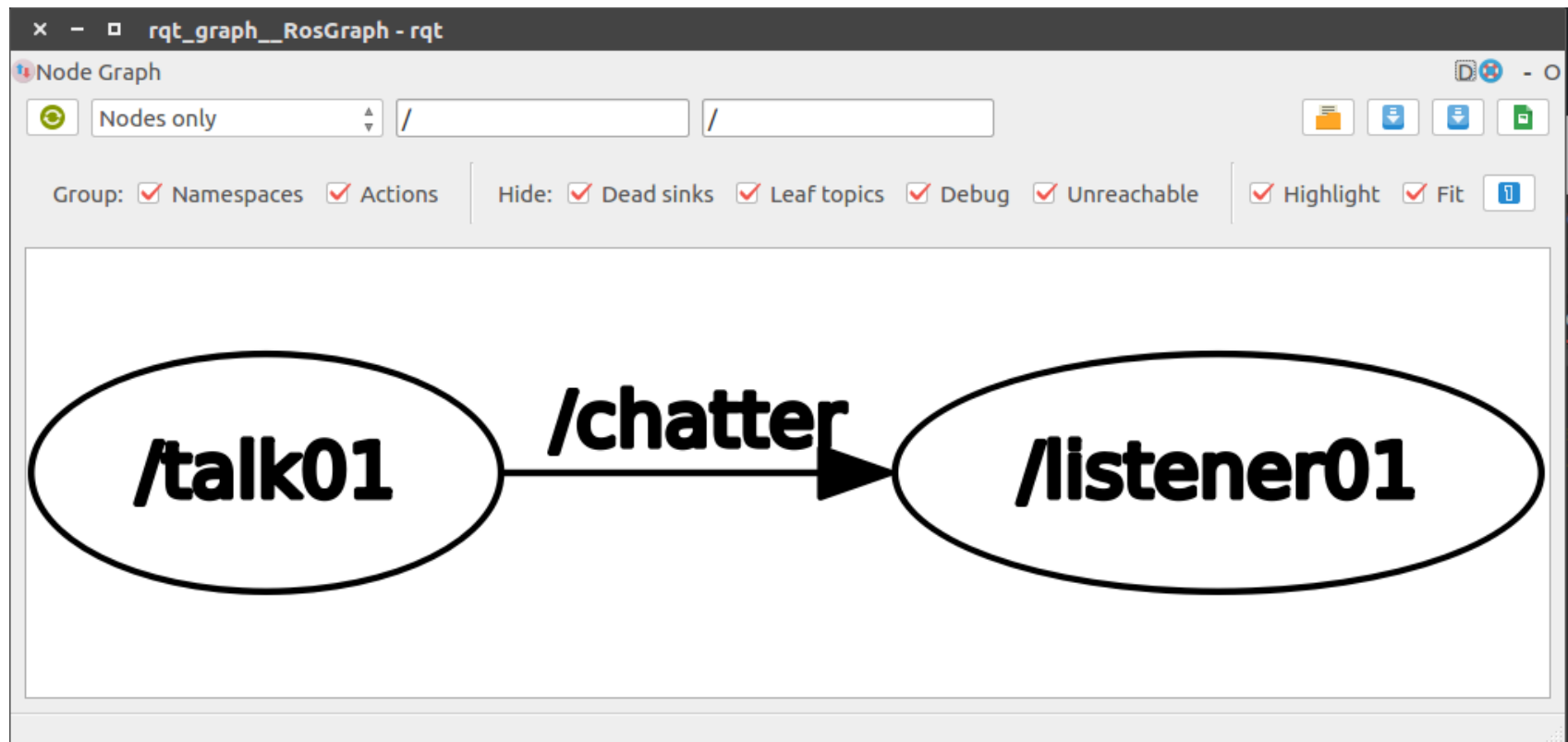
```
x - □ peter@peter-VirtualBox: ~/catkin_ws/src/my_work01/launch
peter@peter-Virtu... x peter@peter-Virtu... x peter@peter-Virtu... x pet
talk01 (my_work01/talk)
auto-starting new master
process[master]: started with pid [8288]
ROS_MASTER_URI=http://localhost:11311
setting /run_id to 15cb136e-96c9-11eb-8823-0800273d7262
process[rosout-1]: started with pid [8301]
started core service [/rosout]
process[talk01-2]: started with pid [8304]
process[listener01-3]: started with pid [8305]
[ INFO] [1617707599.038192056]: hello world 0
[ INFO] [1617707599.138296116]: hello world 1
[ INFO] [1617707599.238236324]: hello world 2
[ INFO] [1617707599.238960639]: I heard: [hello world 2]
[ INFO] [1617707599.338768304]: hello world 3
[ INFO] [1617707599.339505016]: I heard: [hello world 3]
[ INFO] [1617707599.438465194]: hello world 4
[ INFO] [1617707599.438933030]: I heard: [hello world 4]
[ INFO] [1617707599.538319079]: hello world 5
[ INFO] [1617707599.539576146]: I heard: [hello world 5]
[ INFO] [1617707599.638447381]: hello world 6
[ INFO] [1617707599.639036751]: I heard: [hello world 6]
[ INFO] [1617707599.739232131]: hello world 7
<?xml version="1.0"?>
<launch>
  <node pkg="my_work01"
  <node pkg="my_work01"
</launch>
```

```
roslaunch my_work01 talk.launch
rosnode list
rqt_graph
```

```
peter@peter-VirtualBox:~/catkin_ws/src/my_work01/launch$ rosnod list
/listener01
/rosout
/talk01
```

```
<?xml version="1.0"?>
<launch>
  <node pkg="my_work01"
    name="talk01"
    type="talk01" />
  <node pkg="my_work01"
    name="listener01"
    type="listener01" />
</launch>
```

rqt\_graph



## roslaunch 執行多個程式

```
peter@peter-VirtualBox:~/catkin_ws/src/my_work01/launch$ gedit talk.launch
```

```
<?xml version="1.0"?>
<launch>
  <node pkg="my_work01" type="talk" name="talk01" output="screen"/>
  <node pkg="my_work01" type="listener" name="listener01" output="screen"/>
</launch>
```

```
<?xml version="1.0"?>
```

```
<launch>
```

```
  <node pkg="my_work01" type="talk" name="talk01" output="screen"/>
```

```
  <node pkg="my_work01" type="listener" name="listener01" output="screen"/>
```

```
</launch>
```

```
peter@peter-VirtualBox:~/catkin_ws/src/my_work01/launch$ roslaunch my_work01 talk.launch
```

```
x - □ peter@peter-VirtualBox: ~/catkin_ws/src/my_work01/launch
peter@peter-Virtu... x peter@peter-Virtu... x peter@peter-Virtu... x pet
talk01 (my_work01/talk)
auto-starting new master
process[master]: started with pid [8288]
ROS_MASTER_URI=http://localhost:11311
setting /run_id to 15cb136e-96c9-11eb-8823-0800273d7262
process[rosout-1]: started with pid [8301]
started core service [/rosout]
process[talk01-2]: started with pid [8304]
process[listener01-3]: started with pid [8305]
[ INFO] [1617707599.038192056]: hello world 0
[ INFO] [1617707599.138296116]: hello world 1
[ INFO] [1617707599.238236324]: hello world 2
[ INFO] [1617707599.338265381]: hello world 3
```

## roslaunch 執行多個roslaunch

```
peter@peter-lenovo-g50-80:/$ cd ~/catkin_ws/src/my_work01/launch/
```

```
peter@peter-lenovo-g50-80:~/catkin_ws/src/my_work01/launch$ gedit amcl.launch
```

```
<?xml version="1.0"?>
<launch>
  <include file="$(find turtlebot3_gazebo)/launch/turtlebot3_world.launch"/>
  <include file="$(find turtlebot3_navigation)/launch/turtlebot3_navigation.launch"/>
</launch>
```

```
<?xml version="1.0"?>
```

```
<launch>
```

```
  <include file="$(find turtlebot3_gazebo)/launch/turtlebot3_world.launch"/>
```

```
  <include file="$(find turtlebot3_navigation)/launch/turtlebot3_navigation.launch"/>
```

```
</launch>
```

## 執行

```
peter@peter-lenovo-g50-80:~/catkin_ws/src/my_work01/launch$ cd
peter@peter-lenovo-g50-80:~$ export TURTLEBOT3_MODEL=waffle
peter@peter-lenovo-g50-80:~$ roslaunch my_work01 amcl.launch
```