

ROS Message, Service, Param(1)

학과 : 산업인공지능학과
학번 : 2020254016
이름 : 박 민 우

1. ROS Message

1) Create msg

- testpkg에 새로운 폴더 생성, testmsg.msg 파일 생성

```
ubuntu@Park: ~/test_ws/src/testpkg
ubuntu@Park:~/test_ws$ cd test_ws
ubuntu@Park:~/test_ws$ cd src/testpkg
ubuntu@Park:~/test_ws/src/testpkg$ ls
CMakeLists.txt  include  package.xml  src
ubuntu@Park:~/test_ws/src/testpkg$ mkdir msg
ubuntu@Park:~/test_ws/src/testpkg$ touch msg/testmsg.msg
ubuntu@Park:~/test_ws/src/testpkg$ nano msg/testmsg.msg
ubuntu@Park:~/test_ws/src/testpkg$
```

- testmsg.msg 내용 작성

```
GNU nano 2.5.3      파일: msg/testmsg.msg      변경됨.

Header header
int32 x
int32 y
int32[] testarray

^G 도움말 보기      ^O Write Out      ^W 검색      ^K 문자열 잘라내기  ^J 양쪽 정렬      ^C 커서 위치
^X 끝내기          ^R 파일 읽기      ^\ 바꾸기      ^U Uncut Text      ^T 맞춤법          ^_ 행으로 이동
```

- package.xml 파일에 내용 추가

```
*package.xml (~/.test_ws/src/testpkg) - gedit

<exec_depend>roscpp</exec_depend>
<exec_depend>rospy</exec_depend>
<exec_depend>std_msgs</exec_depend>

<build_depend>message_generation</build_depend>
<build_export_depend>message_generation</build_export_depend>
<exec_depend>message_generation</exec_depend>

<!-- The export tag contains other, unspecified, tags -->
<export>
  <!-- Other tools can request additional information be placed here -->
</export>
</package>
```

- CMakeList.txt find_package를 수정

```
CMakeLists.txt (~/.test_ws/src/testpkg) - gedit

cmake_minimum_required(VERSION 3.0.2)
project(testpkg)

## Compile as C++11, supported in ROS Kinetic and newer
# add_compile_options(-std=c++11)

## Find catkin macros and libraries
## if COMPONENTS list like find_package(catkin REQUIRED COMPONENTS xyz)
## is used, also find other catkin packages
find_package(catkin REQUIRED COMPONENTS
  roscpp
  rospy
  std_msgs
  message_generation
)

##
```

- add_message_files 부분을 주석처리 해제하고 수정

```
*CMakeLists.txt (~/.test_ws/src/testpkg) - gedit

##
# * add every package in MSG_DEP_SET to generate_messages(DEPENDENCIES ...)

## Generate messages in the 'msg' folder
add_message_files(
  FILES
  testmsg.msg
)

## Generate services in the 'srv' folder
```

- generate_messages 부분을 주석처리 해제하고 수정

```

# )

## Generate added messages and services with any dependencies listed here
generate_messages(
  DEPENDENCIES
    std_msgs
)

#####
  
```

2) Use custom msg

- custommsg.cpp와 custommsg_sub.cpp 생성

```

#include "ros/ros.h"
#include "testpkg/testmsg.h"
#include <stdlib.h>
#include <iostream>
#include <vector>
std::vector<int> storedVector;
int main(int argc, char **argv)
{
  ros::init(argc, argv, "custommsg_pub");
  ros::NodeHandle n;
  ros::Publisher chatter_pub = n.advertise<testpkg::testmsg>("custommsg", 1);
  ros::Rate loop_rate(10);
  int count = 0;
  for(int i = 0; i < 5; i++)
    storedVector.push_back(i);
  while (ros::ok())
  {
    testpkg::testmsg pub_data;
    pub_data.header.frame_id = "/map";
    pub_data.header.seq = count;
    pub_data.header.stamp = ros::Time::now();
    pub_data.x = 10;
    pub_data.y = 20;
    storedVector.push_back(count*2);
    storedVector.erase(storedVector.begin());
    pub_data.testarray = storedVector;
    chatter_pub.publish(pub_data);
    std::cout << "pub!" << std::endl;
    ros::spinOnce();
    loop_rate.sleep();
    ++count;
  }
  return 0;
}
  
```

```

#include "ros/ros.h"
#include "testpkg/testmsg.h"
#include <iostream>
void msgCallback(const testpkg::testmsg::ConstPtr& msg)
{
  std::cout << "callback!\n";
  std::cout << "msg->header.frame_id " << msg->header.frame_id << "\n";
  std::cout << "msg->header.seq " << msg->header.seq << "\n";
  std::cout << "msg->header.stamp " << msg->header.stamp << "\n";
  std::cout << "msg->x = " << msg->x << "\n";
  std::cout << "msg->y = " << msg->y << "\n";
  std::cout << "msg->testarray.push_back ";
  for(int i = 0; i < msg->testarray.size(); i++)
  {
    std::cout << " " << msg->testarray.at(i);
  }
  std::cout << std::endl;
}

int main(int argc, char **argv)
{
  ros::init(argc, argv, "custommsg_sub");
  ros::NodeHandle n;
  ros::Subscriber sub = n.subscribe("custommsg", 1000, msgCallback);

  ros::spin();

  return 0;
}
  
```

- CMakeList.txt에 노드 추가

```

add_executable(testnode src/testnode.cpp)
target_link_libraries(testnode ${catkin_LIBRARIES})

add_executable(talkernode src/talker.cpp)
target_link_libraries(talkernode ${catkin_LIBRARIES})

add_executable(listenernode src/listener.cpp)
target_link_libraries(listenernode ${catkin_LIBRARIES})

add_executable(custommsgnode msg/custommsg.cpp)
target_link_libraries(custommsgnode ${catkin_LIBRARIES})

add_executable(custommsg_subnode msg/custommsg_sub.cpp)
target_link_libraries(custommsg_subnode ${catkin_LIBRARIES})
  
```

3) 실행 결과

- catkin_make 빌드

```

ubuntu@Park: ~/test_ws
ubuntu@Park:~$ cd test_ws
ubuntu@Park:~/test_ws$ catkin_make
Base path: /home/ubuntu/test_ws
Source space: /home/ubuntu/test_ws/src
Build space: /home/ubuntu/test_ws/build
Devel space: /home/ubuntu/test_ws/devel
Install space: /home/ubuntu/test_ws/install
####
### Running command: "make cmake_check_build_system" in "/home/ubuntu/test_ws/build"
###
-- Using CATKIN_DEVEL_PREFIX: /home/ubuntu/test_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/kinetic
-- This workspace overlays: /opt/ros/kinetic
-- Found PythonInterp: /usr/bin/python2 (found suitable version "2.7.12", minimum required is "2")
-- Using PYTHON_EXECUTABLE: /usr/bin/python2
-- Using Debian Python package layout

```

```

ubuntu@Park: ~/test_ws
[ 35%] Built target _testpkg_generate_messages_check_deps_testmsg
[ 35%] Built target std_msgs_generate_messages_nodejs
[ 35%] Built target std_msgs_generate_messages_cpp
[ 41%] Building CXX object testpkg/CMakeFiles/custommsgnode.dir/msg/custommsg.cpp.o
[ 47%] Building CXX object testpkg/CMakeFiles/custommsg_subnode.dir/msg/custommsg_sub.cpp.o
[ 52%] Built target testpkg_generate_messages_lisp
[ 64%] Built target testpkg_generate_messages_eus
[ 76%] Built target testpkg_generate_messages_py
[ 82%] Built target testpkg_generate_messages_cpp
[ 88%] Built target testpkg_generate_messages_nodejs
[ 88%] Built target testpkg_generate_messages
[ 94%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/custommsgnode
[ 94%] Built target custommsgnode
[100%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/custommsg_subnode
[100%] Built target custommsg_subnode
ubuntu@Park:~/test_ws$

```

- Node 실행

```

ubuntu@Park: ~/test_ws
[ 82%] Built target testpkg_generate_messages_nodejs
[100%] Built target testpkg_generate_messages_lisp
[100%] Built target testpkg_generate_messages_py
[100%] Built target testpkg_generate_messages
ubuntu@Park:~/test_ws$ source devel/setup.sh
ubuntu@Park:~/test_ws$ rosrn testpkg custommsgnode
pub!
pub!
pub!
pub!
pub!
pub!
pub!
pub!
pub!
pub!

```

```

ubuntu@Park: ~/test_ws
[100%] Built target testpkg_generate_messages
ubuntu@Park:~/test_ws$ source devel/setup.sh
ubuntu@Park:~/test_ws$ rosrn testpkg custommsg_subnode
callback!
msg->header.frame_id /map
msg->header.seq 489
msg->header.stamp 1606051827.649858293
msg->x = 10
msg->y = 20
msg->testarray.push_back 970 972 974 976 978
callback!
msg->header.frame_id /map
msg->header.seq 490
msg->header.stamp 1606051827.749858548
msg->x = 10
msg->y = 20
msg->testarray.push_back 972 974 976 978 980
callback!
msg->header.frame_id /map
msg->header.seq 491
msg->header.stamp 1606051827.849856155
msg->x = 10
msg->y = 20
msg->testarray.push_back 974 976 978 980 982
callback!
msg->header.frame_id /map
msg->header.seq 492
msg->header.stamp 1606051827.949859811

```