

### 1- Create a work space;

```
mkdir -p catkin_ws/src
cd catkin_ws/
catkin_make -> build
**source devel/setup.bash
**vim ~/.bashrc
**/source home/ahmet/catkin_ws/devel/setup.bash
```

### 2- Create a package;

```
cd src/
**catkin_create_pkg package_name roscpp rospy std_msgs
catkin_make (cd /home/ahmet/catkin_ws)
```

### 3- Create msg folder

```
/home/ahmet/catkin_ws/src/package_name/msg
**dizine msg dosyalarını oluşturup kaydedin.
**package.xml dosyasından dependencies ler tanımlanır;
***<exec_depend>message_runtime</exec_depend>
***<build_depend>message_generation</build_depend>
**CMakeLists.txt dosyasında düzenlemeler yapılır;
***find_package + message_generation
***add_message_files + name of msg files
***generate_messages -> only activate
***catkin_package -> activate and + message_runtime
http://docs.ros.org/en/jade/api/catkin/html/howto/format2/building\_msgs.html
```

```
catkin_make
```

### 4- Publisher;

```
/home/ahmet/catkin_ws/src/package_name/src
touch publisher.cpp or create with vim
```

```
code;
```

```
#include "ros/ros.h"
```

```
#include "first_ros_package/zaman.h"
```

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

```
{
```

```
    ros::init(argc,argv, "publisher"); // Başlatacağımız node
```

```
    ros::NodeHandle nh; //nh isimli obje, nodeler arasındaki haberleşme için
```

```
    ros::Publisher obje; topic = nh.advertise<first_ros_package::zaman>("msg_time",100); // topic
```

```
oluşturma (publisher nodelerde topic advertise ile tanımlanılır)
```

```
// msg_time ->topic name <package_name::msg_file_name>
```

```
    ros::Rate loop_rate(10);
```

```
    first_ros_package::zaman msg; //zaman-> msg dosyası, msg->obje
```

```
    int count=0;
```

```
    while (ros::ok())
```

```
    {
```

```
        msg.stamp=ros::Time::now();
```

```
        msg.data=count;
```

```

        ROS_INFO("gönderilen mesaj = %d",msg.stamp.sec);
        ROS_INFO("gönderilen mesaj = %d",msg.data);

        objetopic.publish(msg); // yayın yapma

        loop_rate.sleep();
        ++count;

    }

    return 0;

}

```

## 5- Subscriber;

/home/ahmet/catkin\_ws/src/package\_name/src  
touch subscriber.cpp or create with vim

```

#include "ros/ros.h"
#include "first_ros_package/zaman.h"

void func(const first_ros_package::zaman::ConstPtr&msg){

    ROS_INFO("alınan mesaj = %d",msg->stamp.sec);
    ROS_INFO("alınan mesaj = %d",msg->stamp.nsec);
    ROS_INFO("alınan mesaj = %d",msg->data);
}

int main(int argc, char **argv)
{
    ros::init(argc,argv,"subscriber");
    ros::NodeHandle nh;
    ros::Subscriber objetopic=nh.subscribe("msg_time",100,func);
    // fonsiyon içerisinde fonksiyon olduğunda callback oluyor bunun için ros::spin
    ros::spin();

    return 0;
}

```

## 6- Paket Yapılandırması

```

**CMakeLists.txt dosyasında düzenlemeler yapılır;
***find_package + genmsg
***include_directories altındaki includeyi aktif et

***en alta satır ekle
****add_executable(publisher src/publisher.cpp)
****target_link_libraries(publisher ${catkin_LIBRARIES})
****add_dependencies(publisher first_ros_package_generate_messages_cpp)

```

```
****add_executable(subscriber src/subscriber.cpp)
****target_link_libraries(subscriber ${catkin_LIBRARIES})
****add_dependencies(subscriber first_ros_package_generate_messages_cpp)
```

## 6- Create msg folder

/home/ahmet/catkin\_ws/src/package\_name/msg

\*\*dizine msg dosyalarını oluşturup kaydedin.

\*\*package.xml dosyasından dependencies ler tanımlanır;

\*\*\*<exec\_depend>message\_runtime</exec\_depend>

\*\*\*<build\_depend>message\_generation</build\_depend>

\*\*CMakeLists.txt dosyasında düzenlemeler yapılır;

\*\*\*find\_package + message\_generation

\*\*\*add\_message\_files + name of msg files

\*\*\*generate\_messages -> only activate

\*\*\*catkin\_package -> activate and + message\_runtime

[http://docs.ros.org/en/jade/api/catkin/html/howto/format2/building\\_msgs.html](http://docs.ros.org/en/jade/api/catkin/html/howto/format2/building_msgs.html)