



WHEELTEC
轮 趣 科 技

ROS话题 讲解

ROS话题通信机制：



一种一对多、异步的通信机制。

topic命令行指令：

rostopic echo : 打印话题信息

rostopic hz : 话题频率

rostopic info : 话题信息

rostopic list : 列举话题

rostopic pub : 往话题输入信息

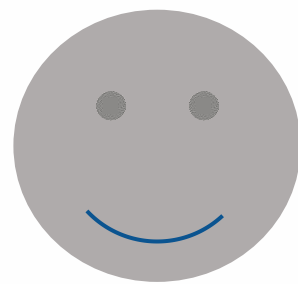
rostopic bw : 话题带宽

rostopic find : 从数据类型寻找话题

rostopic type : 查看话题的数据类型

发布话题：

- ①创建一个发布者
- ②创建消息数据
- ③消息赋值
- ④将消息发布出去



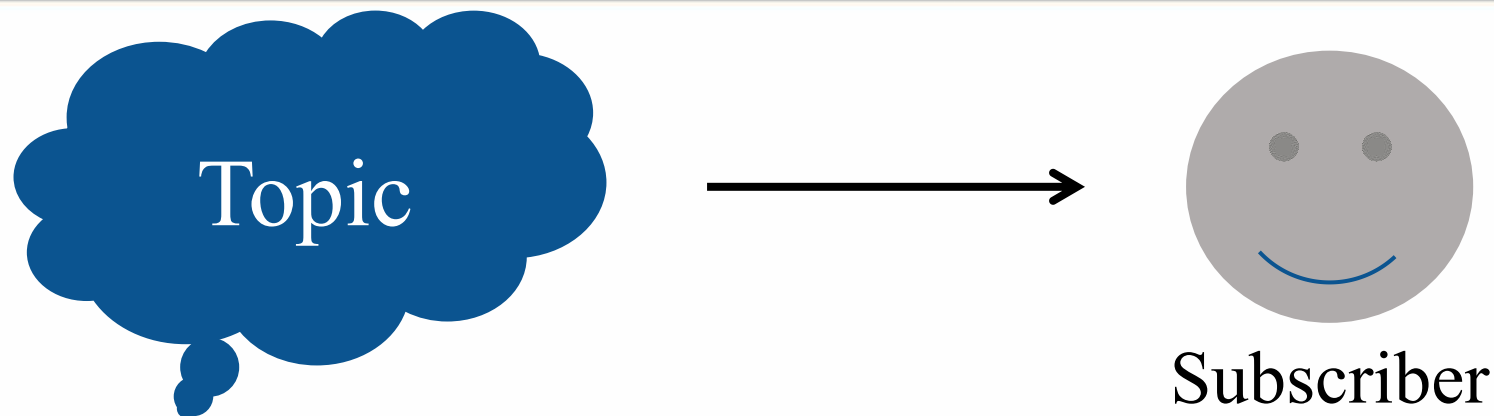
Publisher



```
ros::Publisher pub = nh.advertise<std_msgs::String>("topic_name", queue size);  
std_msgs::String str;  
str.data = "hello world";  
pub.publish(str);
```

订阅话题：

- ①创建一个订阅者
- ②创建一个回调函数



```
void callback(const std_msgs::StringConstPtr& str)
{
    ...
}
...
ros::Subscriber sub = nh.subscribe("my_topic", queue size, callback);
```

自定义消息类型数据：



Topic



rosmmsg list

rosmmsg show

rostopic type /topic_name | rosmmsg show

①自定义msg文件

②添加编译规则

③添加描述规则

自定义消息类型数据：



Topic



package.xml:

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

CMakeLists.txt:

```
find_package (catkin REQUIRED COMPONENTS roscpp std_msgs message_generation)  
add_message_files( FILES Message1.msg )  
generate_messages( DEPENDENCIES std_msgs )  
catkin_package( ... CATKIN_DEPENDS message_runtime ... )
```



WHEELTEC
轮 趣 科 技

THANK YOU

感谢聆听