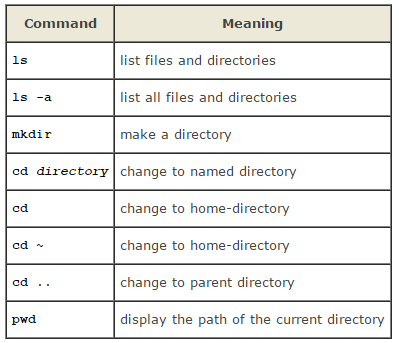
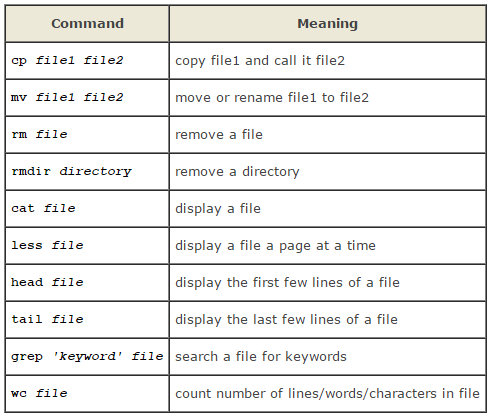
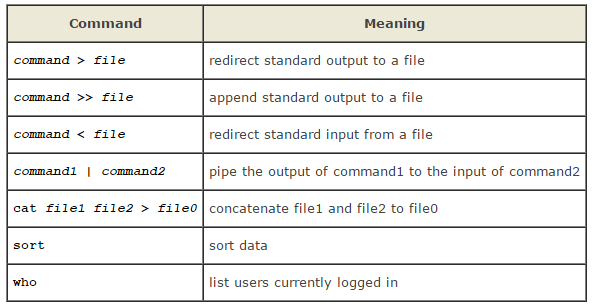
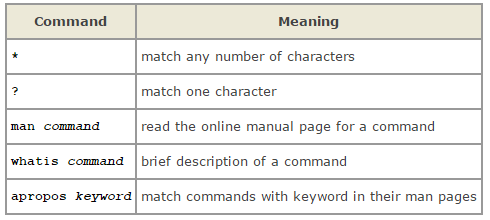
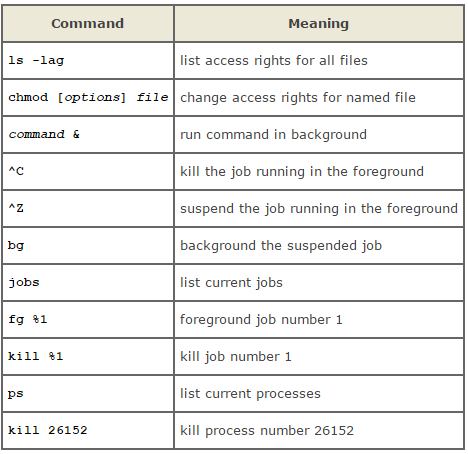
# Unix命令











Ros

## 文件

 Rospack # rospack find [包名称]

**Roscd** # roscd [本地包名称[/子目录]]

**Rosls** # rosls [本地包名称[/子目录]]

Rosmake

rospack depends1/depends [包名称]：查看一级/所有 依赖

## 安装ROS

在开始学习这些教程之前请先按照[ROS安装说明](http://wiki.ros.org/ROS/Installation)完成安装。

**注意:** 如果你是使用类似apt这样的软件管理器来安装ROS的，那么安装后这些软件包将不具备写入权限，当前系统用户比如你自己也无法对这些软件包进行修改编辑。当你的开发涉及到ROS软件包源码层面的操作或者在创建一个新的ROS软件包时，你应该是在一个具备读写权限的目录下工作，就像在你当前系统用户的home目录下一样。

## 管理环境

在安装ROS期间，你会看到提示说需要 source 多个setup.\*sh文件中的某一个，或者甚至提示添加这条'source'命令到你的启动脚本里面。这些操作是必须的，因为ROS是依赖于某种组合空间的概念，而这种概念就是通过配置脚本环境来实现的。这可以让针对不同版本或者不同软件包集的开发更加容易。

如果你在查找和使用ROS软件包方面遇到了问题，请确保你已经正确配置了脚本环境。一个检查的好方法是确保你已经设置了像[ROS\_ROOT](http://wiki.ros.org/ROS/EnvironmentVariables#ROS_ROOT)和[ROS\_PACKAGE\_PATH](http://wiki.ros.org/ROS/EnvironmentVariables#ROS_PACKAGE_PATH)这样的[环境变量](http://wiki.ros.org/ROS/EnvironmentVariables)，可以通过以下命令查看：

$ export | grep ROS

如果发现没有配置，那这个时候你就需要'source'某些'setup.\*sh’文件了。

ROS会帮你自动生成这些‘setup.\*sh’文件，通过以下方式生成并保存在不同地方：

* 通过类似apt的软件包管理器安装ROS软件包时会生成setup.\*sh文件。
* 在[rosbuild workspaces](http://www.ros.org/wiki/fuerte/Installation/Overlays)中通过类似[rosws](http://wiki.ros.org/rosws)的工具生成。
* 在[编译](http://wiki.ros.org/catkin/workspaces#Building_Packages_with_catkin) 或 [安装](http://wiki.ros.org/catkin/workspaces#Installing_Packages_with_Catkin) catkin 软件包时自动生成。

**注意：** 在所有教程中你将会经常看到分别针对[rosbuild](http://wiki.ros.org/rosbuild) 和 [catkin](http://wiki.ros.org/catkin)的不同操作说明，这是因为目前有两种不同的方法可以用来组织和编译ROS应用程序。一般而言，rosbuild比较简单也易于使用，而catkin使用了更加标准的CMake规则，所以比较复杂，但是也更加灵活，特别是对于那些想整合外部现有代码或者想发布自己代码的人。关于这些如果你想了解得更全面请参阅[catkin or rosbuild](http://wiki.ros.org/catkin_or_rosbuild)。

如果你是通过ubuntu上的 apt 工具来安装ROS的，那么你将会在'/opt/ros/<distro>/'目录中看到setup.\*sh文件，然后你可以执行下面的source命令：

# source /opt/ros/<distro>/setup.bash

请使用具体的ROS发行版名称代替<distro>。

比如你安装的是ROS Indigo，则上述命令改为：

$ source /opt/ros/indigo/setup.bash

在每次打开终端时你都需要先运行上面这条命令后才能运行ros相关的命令，为了避免这一繁琐过程，你可以事先在.bashrc文件（初学者请注意：该文件是在当前系统用户的home目录下。）中添加这条命令，这样当你每次登录后系统已经帮你执行这些命令配置好环境。这样做也可以方便你在同一台计算机上安装并随时切换到不同版本的ROS（比如fuerte和groovy）。

此外，你也可以在其它系统平台上相应的ROS安装目录下找到这些setup.\*sh文件。

## 创建ROS的workspace

catkinrosbuild

这些操作方法只适用于ROS Groovy及后期版本，对于ROS Fuerte及早期版本请选择rosbuild。

下面我们开始创建一个[catkin 工作空间](http://wiki.ros.org/catkin/workspaces)：

$ mkdir -p ~/catkin\_ws/src

$ cd ~/catkin\_ws/src

即使这个工作空间是空的（在'src'目录中没有任何软件包，只有一个CMakeLists.txt链接文件），你依然可以编译它：

$ cd ~/catkin\_ws/

$ catkin\_make

[catkin\_make](http://wiki.ros.org/catkin/commands/catkin_make)命令在[catkin 工作空间](http://wiki.ros.org/catkin/workspaces)中是一个非常方便的工具。如果你查看一下当前目录应该能看到'build'和'devel'这两个文件夹。在'devel'文件夹里面你可以看到几个setup.\*sh文件。source这些文件中的任何一个都可以将当前工作空间设置在ROS工作环境的最顶层，想了解更多请参考[catkin](http://wiki.ros.org/catkin)文档。接下来首先source一下新生成的setup.\*sh文件：

$ source devel/setup.bash

要想保证工作空间已配置正确需确保ROS\_PACKAGE\_PATH环境变量包含你的工作空间目录，采用以下命令查看：

$ echo $ROS\_PACKAGE\_PATH

/home/<youruser>/catkin\_ws/src:/opt/ros/indigo/share:/opt/ros/indigo/stacks

到此你的工作环境已经搭建完成，接下来可以继续学习 [ROS文件系统教程](http://wiki.ros.org/cn/ROS/Tutorials/NavigatingTheFilesystem).

## 创建ROS的PACKAGE

首先切换到之前通过[创建catkin工作空间教程](http://wiki.ros.org/catkin/Tutorials/create_a_workspace)创建的catkin工作空间中的src目录下：

# You should have created this in the Creating a Workspace Tutorial

$ cd ~/catkin\_ws/src

现在使用catkin\_create\_pkg命令来创建一个名为'beginner\_tutorials'的新程序包，这个程序包依赖于std\_msgs、roscpp和rospy：

$ catkin\_create\_pkg beginner\_tutorials std\_msgs rospy roscpp

这将会创建一个名为beginner\_tutorials的文件夹，这个文件夹里面包含一个[package.xml](http://wiki.ros.org/catkin/package.xml)文件和一个[CMakeLists.txt](http://wiki.ros.org/catkin/CMakeLists.txt)文件，这两个文件都已经自动包含了部分你在执行catkin\_create\_pkg命令时提供的信息。

catkin\_create\_pkg命令会要求你输入package\_name，如果有需要你还可以在后面添加一些需要依赖的其它程序包：

# This is an example, do not try to run this

# catkin\_create\_pkg <package\_name> [depend1] [depend2] [depend3]

catkin\_create\_pkg命令也有更多的高级功能，这些功能在[catkin/commands/catkin\_create\_pkg](http://wiki.ros.org/catkin/commands/catkin_create_pkg)中有描述。

# 编译ROS的PACKAGE

记得事先source你的环境配置(setup)文件，在Ubuntu中的操作指令如下：

$ source /opt/ros/groovy/setup.bash

[catkin\_make](http://wiki.ros.org/catkin/commands/catkin_make) 是一个命令行工具，它简化了catkin的标准工作流程。你可以认为[catkin\_make](http://wiki.ros.org/catkin/commands/catkin_make)是在CMake标准工作流程中依次调用了cmake 和 make。

使用方法:

# 在catkin工作空间下

$ catkin\_make [make\_targets] [-DCMAKE\_VARIABLES=...]

CMake标准工作流程主要可以分为以下几个步骤：

**注意:** 如果你运行以下命令是无效的，因为它只是一个演示CMake工作流程的例子。

# 在一个CMake项目里

$ mkdir build

$ cd build

$ cmake ..

$ make

$ make install # (可选)

每个CMake工程在编译时都会执行这个操作过程。相反，多个catkin项目可以放在工作空间中一起编译，工作流程如下：

# In a catkin workspace

$ catkin\_make

$ catkin\_make install # (可选)

上述命令会编译src文件夹下的所有catkin工程。想更深入了解请参考[REP128](http://ros.org/reps/rep-0128.html)。 如果你的源代码不在默认工作空间中（~/catkin\_ws/src),比如说存放在了my\_src中，那么你可以这样来使用catkin\_make:

**注意:** 运行以下命令时无效的，因为my\_src不存在。

# In a catkin workspace

$ catkin\_make --source my\_src

$ catkin\_make install --source my\_src # (optionally)

对于[catkin\_make](http://wiki.ros.org/catkin/commands/catkin_make)更高级的使用方法，请参考[catkin/commands/catkin\_make](http://wiki.ros.org/catkin/commands/catkin_make)

# 指定环境变量ROS\_PACKAGE\_PATH

export ROS\_PACKAGE\_PATH=<path> 重新指定

或者source ~/[workspace]/devel/setup.bash

echo $ROS\_PACKAGE\_PATH 查看

# 关闭ROS的MASTER

killall -9 rosmaster

# 更新互联网软件包

 sudo apt-get update更新软件源

 sudo apt-get dist-upgrade,更新所有的软件

# rosnode

rosnode list：看有哪些节点在运行

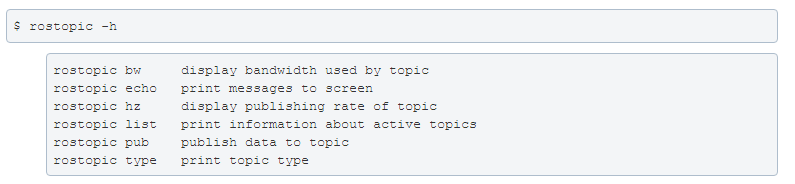
rosnode info /rosout：看节点信息

rosrun [package\_name] [node\_name]：运行节点

rosnode ping [node\_name]：测试节点

# topic

总览：



rosmsg show [mes]：查看消息的详细情况

分览：

* rostopic echo [topic]：可以显示在某个话题上发布的数据。
* $ rostopic list –h：能够列出所有当前订阅和发布的话题。
* Usage: rostopic list [/topic]
* Options:
* -h, --help show this help message and exit
* -b BAGFILE, --bag=BAGFILE
* list topics in .bag file
* -v, --verbose list full details about each topic
* -p list only publishers
* -s list only subscribers
* rostopic type [topic]：查看所发布话题的消息类型。
* rosmsg show [mes]：查看消息的详细情况
* rostopic pub [topic] [msg\_type] [args]：可以把数据发布到当前某个正在广播的话题上。

示例：

示例（非hydro版）：

$ rostopic pub -1 /turtle1/command\_velocity turtlesim/Velocity -- 2.0 1.8

示例（hydro版)：

$ rostopic pub -1 /turtle1/cmd\_vel geometry\_msgs/Twist -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]'

以上命令会发送一条消息给turtlesim，告诉它以2.0大小的线速度和1.8大小的角速度开始移动。

这是一个非常复杂的例子，因此让我们来详细分析一下其中的每一个参数。

* rostopic pub
  + 这条命令将会发布消息到某个给定的话题。
* -1
  + （单个破折号）这个参数选项使rostopic发布一条消息后马上退出。
* /turtle1/command\_velocity
  + 这是消息所发布到的话题名称。
* turtlesim/Velocity
  + 这是所发布消息的类型。
* --
* （双破折号）这会告诉命令选项解析器接下来的参数部分**都不是**命令选项。这在参数里面包含有破折号-（比如负号）时是必须要添加的。
* 2.0 1.8
* 正如之前提到的，在一个turtlesim/Velocity消息里面包含有两个浮点型元素：linear和angular。在本例中，2.0是linear的值，1.8是angular的值。这些参数其实是按照YAML语法格式编写的，这在[YAML文档](http://wiki.ros.org/ROS/YAMLCommandLine)中有更多的描述。

你可能已经注意到turtle已经停止移动了。这是因为turtle需要一个稳定的频率为1Hz的命令流来保持移动状态。我们可以使用rostopic pub -r命令来发布一个稳定的命令流（非hydro版）：

$ rostopic pub /turtle1/command\_velocity turtlesim/Velocity -r 1 -- 2.0 -1.8

hydro版：

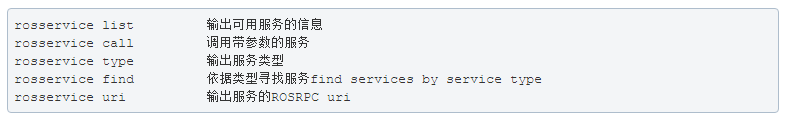
$ rostopic pub /turtle1/cmd\_vel geometry\_msgs/Twist -r 1 -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]'

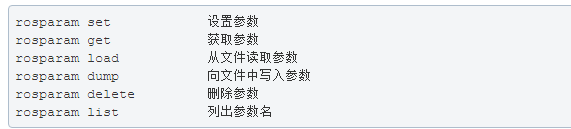
这条命令以1Hz的频率发布速度命令到速度话题上。

* rostopic hz [topic]：可以用来查看数据发布的频率。
* rosrun rqt\_plot rqt\_plot：可以实时显示一个发布到某个话题上的数据变化图形

# Sevice

总览：





分览：

* rosservice list：列出所有服务名称
* rosservice type [service]：查看服务类型

rosservice type [service]| rossrv show：查看带参数的服务和数据

* rosservice call [service] [args]：调用服务

# rosparam

rosparam使得我们能够存储并操作ROS [参数服务器（Parameter Server）](http://wiki.ros.org/%E5%8F%82%E6%95%B0%E6%9C%8D%E5%8A%A1%E5%99%A8%EF%BC%88Parameter%20Server%EF%BC%89)上的数据。参数服务器能够存储整型、浮点、布尔、字符串、字典和列表等数据类型。rosparam使用YAML标记语言的语法。一般而言，YAML的表述很自然：1 是整型, 1.0是浮点型, one是字符串, true是布尔, [1, 2, 3]是整型列表, {a: b, c: d}是字典.

总览：

rosparam set 设置参数

rosparam get 获取参数

rosparam load 从文件读取参数

rosparam dump 向文件中写入参数

rosparam delete 删除参数

rosparam list 列出参数名

分览：

* $ rosparam list：查看参数服务器有哪些参数
* rosparam set [param\_name]：设置参数
* rosparam get [param\_name]：获取参数
* rosparam dump [file\_name]：向文件中写入参数

例如：$ rosparam dump params.yaml

* rosparam load [file\_name] [namespace] :从文件读取参数

例如：

$ rosparam load params.yaml copy

$ rosparam get copy/background\_b

255

# 调试

* $ rosrun rqt\_graph rqt\_graph：查看当前系统运行动态图形
* $ rosrun rqt\_plot rqt\_plot 查看话题数据变化图形
* $ rosrun rqt\_console rqt\_console：显示节点输出信息
* $ rosrun rqt\_logger\_level rqt\_logger\_level：修改节点运行时输出信息的日志等级

# 运行多个节点

$ roslaunch [package] [filename.launch]

# 消息和服务

* [消息(msg)](http://wiki.ros.org/%E6%B6%88%E6%81%AF%28msg%29): msg文件就是一个描述ROS中所使用消息类型的简单文本。它们会被用来生成不同语言的源代码。
* [服务(srv)](http://wiki.ros.org/%E6%9C%8D%E5%8A%A1%28srv%29): 一个srv文件描述一项服务。它包含两个部分：请求和响应。

msg文件存放在package的msg目录下，srv文件则存放在srv目录下。

## 创建消息

下面，我们将在之前创建的package里定义新的消息。

$ cd ~/catkin\_ws/src/beginner\_tutorials

$ mkdir msg

$ echo "int64 num" > msg/Num.msg

上面是最简单的例子——在.msg文件中只有一行数据。当然，你可以仿造上面的形式多增加几行以得到更为复杂的消息：

string first\_name

string last\_name

uint8 age

uint32 score

接下来，还有关键的一步：我们要确保msg文件被转换成为C++，Python和其他语言的源代码：

查看package.xml, 确保它包含一下两条语句:

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

<run\_depend>message\_runtime</run\_depend>

如果没有，添加进去。 注意，在构建的时候，我们只需要"message\_generation"。然而，在运行的时候，我们只需要"message\_runtime"。

在你最喜爱的编辑器中打开CMakeLists.txt文件(可以参考前边的教程[rosed](http://wiki.ros.org/ROS/Tutorials/UsingRosEd)).

在 CMakeLists.txt文件中，利用find\_packag函数，增加对message\_generation的依赖，这样就可以生成消息了。 你可以直接在COMPONENTS的列表里增加message\_generation，就像这样：

# Do not just add this to your CMakeLists.txt, modify the existing text to add message\_generation before the closing parenthesis

find\_package(catkin REQUIRED COMPONENTS

roscpp

rospy

std\_msgs

message\_generation

)

有时候你会发现，即使你没有调用find\_package,你也可以编译通过。这是因为catkin把你所有的package都整合在一起，因此，如果其他的package调用了find\_package，你的package的依赖就会是同样的配置。但是，在你单独编译时，忘记调用find\_package会很容易出错。

同样，你需要确保你设置了运行依赖：

catkin\_package(

...

CATKIN\_DEPENDS message\_runtime ...

...)

找到如下代码块:

# add\_message\_files(

# FILES

# Message1.msg

# Message2.msg

# )

去掉注释符号#，用你的.msg文件替代Message\*.msg，就像下边这样：

add\_message\_files(

FILES

Num.msg

)

By adding the .msg files manually, we make sure that CMake knows when it has to reconfigure the project after you add other .msg files.

Now we must ensure the generate\_messages() function is called.

*For ROS Hydro and later,* you need to uncomment these lines:

# generate\_messages(

# DEPENDENCIES

# std\_msgs

# )

* so it looks like:
* generate\_messages(
* DEPENDENCIES
* std\_msgs

)

*In earlier versions,* you may just need to uncomment one line:

generate\_messages()

使用 rosmsg

以上就是你创建消息的所有步骤。下面通过rosmsg show命令，检查ROS是否能够识消息。

使用方法:

$ rosmsg show [message type]

样例:

$ rosmsg show beginner\_tutorials/Num

你将会看到:

* int64 num

在上边的样例中,消息类型包含两部分：

* beginner\_tutorials -- 消息所在的package
* Num -- 消息名Num.

如果你忘记了消息所在的package，你也可以省略掉package名。输入：

$ rosmsg show Num

你将会看到:

* [beginner\_tutorials/Num]:
* int64 num

There's one more step, though. We need to make sure that the msg files are turned into source code for C++, Python, and other languages. Open CMakeLists.txt in your favorite text editor ([rosed](http://wiki.ros.org/ROS/Tutorials/UsingRosEd) from the previous tutorial is a good option) and remove # to uncomment the following line:

# rosbuild\_genmsg()////（到底需不需要？？？）

## 创建服务

Let's use the package we just created to create a srv:

$ roscd beginner\_tutorials

$ mkdir srv

Instead of creating a new srv definition by hand, we will copy an existing one from another package.

For that, roscp is a useful commandline tool for copying files from one package to another.

Usage:

$ roscp [package\_name] [file\_to\_copy\_path] [copy\_path]

Now we can copy a service from the [rospy\_tutorials](http://wiki.ros.org/rospy_tutorials) package:

$ roscp rospy\_tutorials AddTwoInts.srv srv/AddTwoInts.srv

There's one more step, though. We need to make sure that the srv files are turned into source code for C++, Python, and other languages.

Unless you have done so already, open package.xml, and make sure these two lines are in it and [uncommented](http://www.htmlhelp.com/reference/wilbur/misc/comment.html):

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

<run\_depend>message\_runtime</run\_depend>

As before, note that at build time, we need "message\_generation", while at runtime, we only need "message\_runtime".

Unless you have done so already for messages in the previous step, add the message\_generation dependency to generate messages in CMakeLists.txt:

# Do not just add this line to your CMakeLists.txt, modify the existing line

find\_package(catkin REQUIRED COMPONENTS

roscpp

rospy

std\_msgs

message\_generation

)

(Despite its name, message\_generation works for both msg and srv.)

Also you need the same changes to package.xml for services as for messages, so look above for the additional dependencies required.

Remove # to uncomment the following lines:

# add\_service\_files(

# FILES

# Service1.srv

# Service2.srv

# )

And replace the placeholder Service\*.srv files for your service files:

add\_service\_files(

FILES

AddTwoInts.srv

)

Once again, open CMakeLists.txt and remove # to uncomment the following line:

# rosbuild\_gensrv()////（到底需不需要？？？）

Now you're ready to generate source files from your service definition. If you want to do so right now, skip next sections to[Common step for msg and srv](http://wiki.ros.org/ROS/Tutorials/CreatingMsgAndSrv#Common_step_for_msg_and_srv).

**Using rossrv**

That's all you need to do to create a srv. Let's make sure that ROS can see it using the rossrv show command.

Usage:

$ rossrv show <service type>

Example:

$ rossrv show beginner\_tutorials/AddTwoInts

You will see:

* int64 a
* int64 b
* ---
* int64 sum

## 重新编译

由于增加了新的消息，所以我们需要重新编译我们的package：

# In your catkin workspace

$ cd ../..

$ catkin\_make

$ cd -

所有在msg路径下的.msg文件都将转换为ROS所支持语言的源代码。生成的C++头文件将会放置在~/catkin\_ws/devel/include/beginner\_tutorials/。 Python脚本语言会在 ~/catkin\_ws/devel/lib/python2.7/dist-packages/beginner\_tutorials/msg 目录下创建。 lisp文件会出现在 ~/catkin\_ws/devel/share/common-lisp/ros/beginner\_tutorials/msg/ 路径下.

详尽的消息格式请参考[Message Description Language](http://wiki.ros.org/ROS/Message_Description_Language) 页面.

# 编写发布器和订阅器

编写发布器节点

『节点』(Node) 是指 ROS 网络中可执行文件。接下来，我们将会创建一个发布器节点("talker")，它将不断的在 ROS 网络中广播消息。

切换到之前创建的 beginner\_tutorials package 路径下：

roscd beginner\_tutorials

切换到之前创建的 beginner\_tutorials package 路径下：

cd ~/catkin\_ws/src/beginner\_tutorials

源代码

在 beginner\_tutorials package 路径下创建一个src文件夹：

mkdir -p ~/catkin\_ws/src/beginner\_tutorials/src

这个文件夹将会用来放置 beginner\_tutorials package 的所有源代码。

在 beginner\_tutorials package 里创建 src/talker.cpp 文件，并将如下代码粘贴到文件内：

[*https://raw.github.com/ros/ros\_tutorials/groovy-devel/roscpp\_tutorials/talker/talker.cpp*](https://raw.github.com/ros/ros_tutorials/groovy-devel/roscpp_tutorials/talker/talker.cpp)

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[27](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_27) #include "ros/ros.h"

[28](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_28) #include "std\_msgs/String.h"

[29](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_29)

[30](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_30) #include <sstream>

[31](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_31)

[32](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_32) /\*\*

[33](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_33) \* This tutorial demonstrates simple sending of messages over the ROS system.

[34](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_34) \*/

[35](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_35) int main(int argc, char \*\*argv)

[36](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_36) {

[37](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_37) /\*\*

[38](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_38) \* The ros::init() function needs to see argc and argv so that it can perform

[39](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_39) \* any ROS arguments and name remapping that were provided at the command line. For programmatic

[40](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_40) \* remappings you can use a different version of init() which takes remappings

[41](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_41) \* directly, but for most command-line programs, passing argc and argv is the easiest

[42](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_42) \* way to do it. The third argument to init() is the name of the node.

[43](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_43) \*

[44](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_44) \* You must call one of the versions of ros::init() before using any other

[45](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_45) \* part of the ROS system.

[46](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_46) \*/

[47](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_47) ros::init(argc, argv, "talker");

[48](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_48)

[49](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_49) /\*\*

[50](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_50) \* NodeHandle is the main access point to communications with the ROS system.

[51](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_51) \* The first NodeHandle constructed will fully initialize this node, and the last

[52](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_52) \* NodeHandle destructed will close down the node.

[53](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_53) \*/

[54](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_54) ros::NodeHandle n;

[55](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_55)

[56](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_56) /\*\*

[57](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_57) \* The advertise() function is how you tell ROS that you want to

[58](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_58) \* publish on a given topic name. This invokes a call to the ROS

[59](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_59) \* master node, which keeps a registry of who is publishing and who

[60](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_60) \* is subscribing. After this advertise() call is made, the master

[61](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_61) \* node will notify anyone who is trying to subscribe to this topic name,

[62](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_62) \* and they will in turn negotiate a peer-to-peer connection with this

[63](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_63) \* node. advertise() returns a Publisher object which allows you to

[64](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_64) \* publish messages on that topic through a call to publish(). Once

[65](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_65) \* all copies of the returned Publisher object are destroyed, the topic

[66](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_66) \* will be automatically unadvertised.

[67](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_67) \*

[68](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_68) \* The second parameter to advertise() is the size of the message queue

[69](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_69) \* used for publishing messages. If messages are published more quickly

[70](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_70) \* than we can send them, the number here specifies how many messages to

[71](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_71) \* buffer up before throwing some away.

[72](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_72) \*/

[73](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_73) ros::Publisher chatter\_pub = n.advertise<std\_msgs::String>("chatter", 1000);

[74](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_74)

[75](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_75) ros::Rate loop\_rate(10);

[76](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_76)

[77](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_77) /\*\*

[78](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_78) \* A count of how many messages we have sent. This is used to create

[79](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_79) \* a unique string for each message.

[80](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_80) \*/

[81](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_81) int count = 0;

[82](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_82) while (ros::ok())

[83](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_83) {

[84](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_84) /\*\*

[85](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_85) \* This is a message object. You stuff it with data, and then publish it.

[86](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_86) \*/

[87](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_87) std\_msgs::String msg;

[88](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_88)

[89](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_89) std::stringstream ss;

[90](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_90) ss << "hello world " << count;

[91](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_91) msg.data = ss.str();

[92](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_92)

[93](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_93) ROS\_INFO("%s", msg.data.c\_str());

[94](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_94)

[95](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_95) /\*\*

[96](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_96) \* The publish() function is how you send messages. The parameter

[97](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_97) \* is the message object. The type of this object must agree with the type

[98](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_98) \* given as a template parameter to the advertise<>() call, as was done

[99](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_99) \* in the constructor above.

[100](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_100) \*/

[101](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_101) chatter\_pub.publish(msg);

[102](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_102)

[103](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_103) ros::spinOnce();

[104](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_104)

[105](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_105) loop\_rate.sleep();

[106](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_106) ++count;

[107](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_107) }

[108](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_108)

[109](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_109)

[110](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_110) return 0;

[111](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-2f0de5afcea33b164ff81dfbdb2ca623dfa24042_111) }

代码说明

现在，我们来分段解释代码。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[27](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-8c1080856928989e4b4a21f76a6dbdb47f49196f_27) #include "ros/ros.h"

[28](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-8c1080856928989e4b4a21f76a6dbdb47f49196f_28)

ros/ros.h 是一个实用的头文件，它引用了 ROS 系统中大部分常用的头文件。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[28](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-dcabc2a2d315305bc1e9eb823364cbdade421375_28) #include "std\_msgs/String.h"

[29](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-dcabc2a2d315305bc1e9eb823364cbdade421375_29)

这引用了 [std\_msgs/String](http://docs.ros.org/api/std_msgs/html/msg/String.html) 消息, 它存放在 [std\_msgs](http://wiki.ros.org/std_msgs) package 里，是由 String.msg 文件自动生成的头文件。需要关于消息的定义，可以参考 [msg](http://wiki.ros.org/msg) 页面。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[47](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-bbd7c4e67305a8926e488e6a5fa5695747e525f9_47) ros::init(argc, argv, "talker");

初始化 ROS 。它允许 ROS 通过命令行进行名称重映射——然而这并不是现在讨论的重点。在这里，我们也可以指定节点的名称——运行过程中，节点的名称必须唯一。

这里的名称必须是一个 [base name](http://wiki.ros.org/Names#Graph) ，也就是说，名称内不能包含 / 等符号。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[54](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-1079aa1401a6c4cbfa4d833c391f1312fc5f3ba3_54) ros::NodeHandle n;

为这个进程的节点创建一个句柄。第一个创建的 NodeHandle 会为节点进行初始化，最后一个销毁的 NodeHandle 则会释放该节点所占用的所有资源。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[73](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-48c016747d035bfd32771b254802cfbd5feafe7d_73) ros::Publisher chatter\_pub = n.advertise<std\_msgs::String>("chatter", 1000);

告诉 master 我们将要在 chatter（话题名） 上发布 [std\_msgs/String](http://docs.ros.org/api/std_msgs/html/msg/String.html) 消息类型的消息。这样 master 就会告诉所有订阅了 chatter 话题的节点，将要有数据发布。第二个参数是发布序列的大小。如果我们发布的消息的频率太高，缓冲区中的消息在大于 1000 个的时候就会开始丢弃先前发布的消息。

NodeHandle::advertise() 返回一个 ros::Publisher 对象,它有两个作用： 1) 它有一个 publish() 成员函数可以让你在topic上发布消息； 2) 如果消息类型不对,它会拒绝发布。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[75](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-436a142c5bd0d3fab1d54147bea31450c8941b19_75) ros::Rate loop\_rate(10);

ros::Rate 对象可以允许你指定自循环的频率。它会追踪记录自上一次调用 Rate::sleep() 后时间的流逝，并休眠直到一个频率周期的时间。

在这个例子中，我们让它以 10Hz 的频率运行。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[81](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-0dd0ef99791fdf3e69587d021e2663b3d8181408_81) int count = 0;

[82](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-0dd0ef99791fdf3e69587d021e2663b3d8181408_82) while (ros::ok())

[83](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-0dd0ef99791fdf3e69587d021e2663b3d8181408_83) {

roscpp 会默认生成一个 SIGINT 句柄，它负责处理 Ctrl-C 键盘操作——使得 ros::ok() 返回 false。

如果下列条件之一发生，ros::ok() 返回false：

* SIGINT 被触发 (Ctrl-C)
* 被另一同名节点踢出 ROS 网络
* ros::shutdown() 被程序的另一部分调用
* 节点中的所有 ros::[NodeHandles](http://wiki.ros.org/NodeHandles) 都已经被销毁

一旦 ros::ok() 返回 false, 所有的 ROS 调用都会失效。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[87](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-8456e0bb7d9b8e4831fbf057e53685ec7270519e_87) std\_msgs::String msg;

[88](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-8456e0bb7d9b8e4831fbf057e53685ec7270519e_88)

[89](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-8456e0bb7d9b8e4831fbf057e53685ec7270519e_89) std::stringstream ss;

[90](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-8456e0bb7d9b8e4831fbf057e53685ec7270519e_90) ss << "hello world " << count;

[91](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-8456e0bb7d9b8e4831fbf057e53685ec7270519e_91) msg.data = ss.str();

我们使用一个由 [msg file](http://wiki.ros.org/msg) 文件产生的『消息自适应』类在 ROS 网络中广播消息。现在我们使用标准的String消息，它只有一个数据成员 "data"。当然，你也可以发布更复杂的消息类型。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[101](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-aa202e8d4afb2f5fef5365e963060fa589e97965_101) chatter\_pub.publish(msg);

这里，我们向所有订阅 chatter 话题的节点发送消息。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[93](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-ff7616a808b0d827bbb7a305f4b0a9b18a7d4309_93) ROS\_INFO("%s", msg.data.c\_str());

ROS\_INFO 和其他类似的函数可以用来代替 printf/cout 等函数。具体可以参考 [rosconsole documentation](http://wiki.ros.org/rosconsole)，以获得更多信息。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[103](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-77f59f9b7961ee481f7596923b938df8c19148f6_103) ros::spinOnce();

在这个例子中并不是一定要调用 ros::spinOnce()，因为我们不接受回调。然而，如果你的程序里包含其他回调函数，最好在这里加上 ros::spinOnce()这一语句，否则你的回调函数就永远也不会被调用了。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[105](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-abbdd3c55b5ae72f624a06429120470b29dffbf7_105) loop\_rate.sleep();

这条语句是调用 ros::Rate 对象来休眠一段时间以使得发布频率为 10Hz。

对上边的内容进行一下总结：

* 初始化 ROS 系统
* 在 ROS 网络内广播我们将要在 chatter 话题上发布 [std\_msgs/String](http://docs.ros.org/api/std_msgs/html/msg/String.html) 类型的消息
* 以每秒 10 次的频率在 chatter 上发布消息

接下来我们要编写一个节点来接收这个消息。

编写订阅器节点

源代码

在 beginner\_tutorials package 目录下创建 src/listener.cpp 文件，并粘贴如下代码：

[*https://raw.github.com/ros/ros\_tutorials/groovy-devel/roscpp\_tutorials/listener/listener.cpp*](https://raw.github.com/ros/ros_tutorials/groovy-devel/roscpp_tutorials/listener/listener.cpp)

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[28](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_28) #include "ros/ros.h"

[29](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_29) #include "std\_msgs/String.h"

[30](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_30)

[31](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_31) /\*\*

[32](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_32) \* This tutorial demonstrates simple receipt of messages over the ROS system.

[33](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_33) \*/

[34](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_34) void chatterCallback(const std\_msgs::String::ConstPtr& msg)

[35](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_35) {

[36](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_36) ROS\_INFO("I heard: [%s]", msg->data.c\_str());

[37](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_37) }

[38](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_38)

[39](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_39) int main(int argc, char \*\*argv)

[40](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_40) {

[41](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_41) /\*\*

[42](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_42) \* The ros::init() function needs to see argc and argv so that it can perform

[43](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_43) \* any ROS arguments and name remapping that were provided at the command line. For programmatic

[44](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_44) \* remappings you can use a different version of init() which takes remappings

[45](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_45) \* directly, but for most command-line programs, passing argc and argv is the easiest

[46](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_46) \* way to do it. The third argument to init() is the name of the node.

[47](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_47) \*

[48](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_48) \* You must call one of the versions of ros::init() before using any other

[49](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_49) \* part of the ROS system.

[50](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_50) \*/

[51](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_51) ros::init(argc, argv, "listener");

[52](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_52)

[53](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_53) /\*\*

[54](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_54) \* NodeHandle is the main access point to communications with the ROS system.

[55](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_55) \* The first NodeHandle constructed will fully initialize this node, and the last

[56](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_56) \* NodeHandle destructed will close down the node.

[57](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_57) \*/

[58](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_58) ros::NodeHandle n;

[59](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_59)

[60](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_60) /\*\*

[61](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_61) \* The subscribe() call is how you tell ROS that you want to receive messages

[62](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_62) \* on a given topic. This invokes a call to the ROS

[63](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_63) \* master node, which keeps a registry of who is publishing and who

[64](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_64) \* is subscribing. Messages are passed to a callback function, here

[65](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_65) \* called chatterCallback. subscribe() returns a Subscriber object that you

[66](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_66) \* must hold on to until you want to unsubscribe. When all copies of the Subscriber

[67](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_67) \* object go out of scope, this callback will automatically be unsubscribed from

[68](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_68) \* this topic.

[69](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_69) \*

[70](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_70) \* The second parameter to the subscribe() function is the size of the message

[71](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_71) \* queue. If messages are arriving faster than they are being processed, this

[72](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_72) \* is the number of messages that will be buffered up before beginning to throw

[73](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_73) \* away the oldest ones.

[74](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_74) \*/

[75](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_75) ros::Subscriber sub = n.subscribe("chatter", 1000, chatterCallback);

[76](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_76)

[77](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_77) /\*\*

[78](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_78) \* ros::spin() will enter a loop, pumping callbacks. With this version, all

[79](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_79) \* callbacks will be called from within this thread (the main one). ros::spin()

[80](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_80) \* will exit when Ctrl-C is pressed, or the node is shutdown by the master.

[81](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_81) \*/

[82](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_82) ros::spin();

[83](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_83)

[84](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_84) return 0;

[85](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-3e05b6b19dba6593660b407d3dceef092933b43a_85) }

代码说明

下面我们将逐条解释代码，当然，之前解释过的代码就不再赘述了。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[34](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-6d9609d1db3a7bdca0302c1203d82078afb15323_34) void chatterCallback(const std\_msgs::String::ConstPtr& msg)

[35](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-6d9609d1db3a7bdca0302c1203d82078afb15323_35) {

[36](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-6d9609d1db3a7bdca0302c1203d82078afb15323_36) ROS\_INFO("I heard: [%s]", msg->data.c\_str());

[37](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-6d9609d1db3a7bdca0302c1203d82078afb15323_37) }

这是一个回调函数，当接收到 chatter 话题的时候就会被调用。消息是以 [boost shared\_ptr](http://www.boost.org/doc/libs/1_37_0/libs/smart_ptr/shared_ptr.htm) 指针的形式传输，这就意味着你可以存储它而又不需要复制数据。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[75](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-5796660589beb27dca39a7b53df11813783e220d_75) ros::Subscriber sub = n.subscribe("chatter", 1000, chatterCallback);

告诉 master 我们要订阅 chatter 话题上的消息。当有消息发布到这个话题时，ROS 就会调用 chatterCallback() 函数。第二个参数是队列大小，以防我们处理消息的速度不够快，当缓存达到 1000 条消息后，再有新的消息到来就将开始丢弃先前接收的消息。

NodeHandle::subscribe() 返回 ros::Subscriber 对象,你必须让它处于活动状态直到你不再想订阅该消息。当这个对象销毁时，它将自动退订 chatter 话题的消息。

有各种不同的 NodeHandle::subscribe() 函数，允许你指定类的成员函数，甚至是 Boost.Function 对象可以调用的任何数据类型。[roscpp overview](http://wiki.ros.org/roscpp/Overview) 提供了更为详尽的信息。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[82](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-6424988503448f0a5323c4531325345a7f643fe3_82) ros::spin();

ros::spin() 进入自循环，可以尽可能快的调用消息回调函数。如果没有消息到达，它不会占用很多 CPU，所以不用担心。一旦 ros::ok() 返回 false，ros::spin() 就会立刻跳出自循环。这有可能是 ros::shutdown() 被调用，或者是用户按下了 Ctrl-C，使得 master 告诉节点要终止运行。也有可能是节点被人为关闭的。

还有其他的方法进行回调，但在这里我们不涉及。想要了解，可以参考 [roscpp\_tutorials](http://wiki.ros.org/roscpp_tutorials) package 里的一些 demo 应用。需要更为详尽的信息，可以参考 [roscpp overview](http://wiki.ros.org/roscpp/Overview)。

下边，我们来总结一下:

* 初始化ROS系统
* 订阅 chatter 话题
* 进入自循环，等待消息的到达
* 当消息到达，调用 chatterCallback() 函数

编译节点

[roscreate-pkg](http://wiki.ros.org/roscreate) 会自动为你创建 Makefile 和 CMakeLists.txt 文件

$ rosed beginner\_tutorials CMakeLists.txt

看起来应该像这样:

* cmake\_minimum\_required(VERSION 2.4.6)
* include($ENV{ROS\_ROOT}/core/rosbuild/rosbuild.cmake)
* # Set the build type. Options are:
* # Coverage : w/ debug symbols, w/o optimization, w/ code-coverage
* # Debug : w/ debug symbols, w/o optimization
* # Release : w/o debug symbols, w/ optimization
* # RelWithDebInfo : w/ debug symbols, w/ optimization
* # MinSizeRel : w/o debug symbols, w/ optimization, stripped binaries
* #set(ROS\_BUILD\_TYPE RelWithDebInfo)
* rosbuild\_init()
* #set the default path for built executables to the "bin" directory
* set(EXECUTABLE\_OUTPUT\_PATH ${PROJECT\_SOURCE\_DIR}/bin)
* #set the default path for built libraries to the "lib" directory
* set(LIBRARY\_OUTPUT\_PATH ${PROJECT\_SOURCE\_DIR}/lib)
* #uncomment if you have defined messages
* #rosbuild\_genmsg()
* #uncomment if you have defined services
* #rosbuild\_gensrv()
* #common commands for building c++ executables and libraries
* #rosbuild\_add\_library(${PROJECT\_NAME} src/example.cpp)
* #target\_link\_libraries(${PROJECT\_NAME} another\_library)
* #rosbuild\_add\_boost\_directories()
* #rosbuild\_link\_boost(${PROJECT\_NAME} thread)
* #rosbuild\_add\_executable(example examples/example.cpp)
* #target\_link\_libraries(example ${PROJECT\_NAME})

在最后增加如下语句:

rosbuild\_add\_executable(talker src/talker.cpp)

rosbuild\_add\_executable(listener src/listener.cpp)

这会生成两个可执行文件, talker 和 listener，编译结果默认存放在 "bin" 目录下.

想要了解更多关于 ROS 下 CMake 的信息，可以参考 [CMakeLists](http://wiki.ros.org/CMakeLists) 现在运行make:

$ make

编译节点

之前教程中使用 [catkin\_create\_pkg](http://wiki.ros.org/catkin/commands/catkin_create_pkg) 创建了 [package.xml](http://wiki.ros.org/catkin/package_manifest) 和 [CMakeLists.txt](http://wiki.ros.org/catkin/CMakeLists.txt) 文件。

生成的 [CMakeLists.txt](http://wiki.ros.org/catkin/CMakeLists.txt) 看起来应该是这样(在 [Creating Msgs and Srvs](http://wiki.ros.org/ROS/Tutorials/CreatingMsgAndSrv) 教程中的修改和未被使用的注释和例子都被移除了):

[*https://raw.github.com/ros/catkin\_tutorials/master/create\_package\_modified/catkin\_ws/src/beginner\_tutorials/CMakeLists.txt*](https://raw.github.com/ros/catkin_tutorials/master/create_package_modified/catkin_ws/src/beginner_tutorials/CMakeLists.txt)

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[1](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_1) cmake\_minimum\_required(VERSION 2.8.3)

[2](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_2) project(beginner\_tutorials)

[3](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_3)

[4](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_4) ## Find catkin and any catkin packages

[5](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_5) find\_package(catkin REQUIRED COMPONENTS roscpp rospy std\_msgs genmsg)

[6](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_6)

[7](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_7) ## Declare ROS messages and services

[8](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_8) add\_message\_files(DIRECTORY msg FILES Num.msg)

[9](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_9) add\_service\_files(DIRECTORY srv FILES AddTwoInts.srv)

[10](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_10)

[11](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_11) ## Generate added messages and services

[12](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_12) generate\_messages(DEPENDENCIES std\_msgs)

[13](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_13)

[14](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_14) ## Declare a catkin package

[15](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-613be11e3a734306735eeef98b24f783cdadf938_15) catkin\_package()

在 [CMakeLists.txt](http://wiki.ros.org/catkin/CMakeLists.txt) 文件末尾加入几条语句:

include\_directories(include ${catkin\_INCLUDE\_DIRS})

add\_executable(talker src/talker.cpp)

target\_link\_libraries(talker ${catkin\_LIBRARIES})

add\_executable(listener src/listener.cpp)

target\_link\_libraries(listener ${catkin\_LIBRARIES})

结果，[CMakeLists.txt](http://wiki.ros.org/catkin/CMakeLists.txt) 文件看起来大概是这样:

[*https://raw.github.com/ros/catkin\_tutorials/master/create\_package\_pubsub/catkin\_ws/src/beginner\_tutorials/CMakeLists.txt*](https://raw.github.com/ros/catkin_tutorials/master/create_package_pubsub/catkin_ws/src/beginner_tutorials/CMakeLists.txt)

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++))

[1](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_1) cmake\_minimum\_required(VERSION 2.8.3)

[2](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_2) project(beginner\_tutorials)

[3](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_3)

[4](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_4) ## Find catkin and any catkin packages

[5](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_5) find\_package(catkin REQUIRED COMPONENTS roscpp rospy std\_msgs genmsg)

[6](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_6)

[7](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_7) ## Declare ROS messages and services

[8](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_8) add\_message\_files(FILES Num.msg)

[9](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_9) add\_service\_files(FILES AddTwoInts.srv)

[10](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_10)

[11](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_11) ## Generate added messages and services

[12](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_12) generate\_messages(DEPENDENCIES std\_msgs)

[13](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_13)

[14](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_14) ## Declare a catkin package

[15](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_15) catkin\_package()

[16](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_16)

[17](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_17) ## Build talker and listener

[18](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_18) include\_directories(include ${catkin\_INCLUDE\_DIRS})

[19](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_19)

[20](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_20) add\_executable(talker src/talker.cpp)

[21](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_21) target\_link\_libraries(talker ${catkin\_LIBRARIES})

[22](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_22) add\_dependencies(talker beginner\_tutorials\_generate\_messages\_cpp)

[23](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_23)

[24](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_24) add\_executable(listener src/listener.cpp)

[25](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_25) target\_link\_libraries(listener ${catkin\_LIBRARIES})

[26](http://wiki.ros.org/cn/ROS/Tutorials/WritingPublisherSubscriber(c++)#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingPublisherSubscriber.CA-56e552a9f29027036adb74aeea6375a15de4f70e_26) add\_dependencies(listener beginner\_tutorials\_generate\_messages\_cpp)

这会生成两个可执行文件, talker 和 listener, 默认存储到 [devel space](http://wiki.ros.org/catkin/workspaces#Development_.28Devel.29_Space) 目录下,具体是在~/catkin\_ws/devel/lib/<package name> 中.

现在要为可执行文件添加对生成的消息文件的依赖：

add\_dependencies(talker beginner\_tutorials\_generate\_messages\_cpp)

这样就可以确保自定义消息的头文件在被使用之前已经被生成。因为 catkin 把所有的 package 并行的编译，所以如果你要使用其他 catkin 工作空间中其他 package 的消息，你同样也需要添加对他们各自生成的消息文件的依赖。当然，如果在 \*Groovy\* 版本下，你可以使用下边的这个变量来添加对所有必须的文件依赖:

add\_dependencies(talker ${catkin\_EXPORTED\_TARGETS})

你可以直接调用可执行文件，也可以使用 rosrun 来调用他们。他们不会被安装到 <prefix>/bin 路径下，因为那样会改变系统的 PATH 环境变量。如果你确定要将可执行文件安装到该路径下，你需要设置安装位置，请参考 [catkin/CMakeLists.txt](http://wiki.ros.org/catkin/CMakeLists.txt)

如果需要关于 [CMakeLists.txt](http://wiki.ros.org/catkin/CMakeLists.txt) 更详细的信息，请参考 [catkin/CMakeLists.txt](http://wiki.ros.org/catkin/CMakeLists.txt)

现在运行 catkin\_make：

# In your catkin workspace

$ catkin\_make

注意：如果你是添加了新的 package，你需要通过 --force-cmake 选项告诉 catkin 进行强制编译。参考 [catkin/Tutorials/using\_a\_workspace#With\_catkin\_make](http://wiki.ros.org/catkin/Tutorials/using_a_workspace#With_catkin_make)。

既然已经编写好了发布器和订阅器，下面让我们来[测试消息发布器和订阅器](http://wiki.ros.org/cn/ROS/Tutorials/ExaminingPublisherSubscriber)。

# 测试发布器和订阅器

确保roscore可用，并运行：

$ roscore

**catkin specific** 如果使用catkin，确保你在调用catkin\_make后，在运行你自己的程序前，已经source了catkin工作空间下的setup.sh文件：

# In your catkin workspace

$ cd ~/catkin\_ws

$ source ./devel/setup.bash

In the last tutorial we made a publisher called "talker". Let's run it:

$ rosrun beginner\_tutorials talker (C++)

$ rosrun beginner\_tutorials talker.py (Python)

# 编写服务器和客户端

编写Service节点

这里，我们将创建一个简单的service节点("add\_two\_ints\_server")，该节点将接收到两个整形数字，并返回它们的和。

进入先前你在[creating a rosbuild package](http://wiki.ros.org/ROS/Tutorials/CreatingPackage)教程中所创建的beginner\_tutorials包所在的目录：

roscd beginner\_tutorials

进入先前你在catkin workspace教程中所创建的beginner\_tutorials包所在的目录：

cd ~/catkin\_ws/src/beginner\_tutorials

请确保已经按照[creating the AddTwoInts.srv](http://wiki.ros.org/ROS/Tutorials/CreatingMsgAndSrv#Creating_a_srv)教程的步骤创建了本教程所需要的srv（确保选择了对应的编译系统“catkin”和“rosbuild”）。

代码

在beginner\_tutorials包中创建src/add\_two\_ints\_server.cpp文件，并复制粘贴下面的代码：

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29)

[1](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_1) #include "ros/ros.h"

[2](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_2) #include "beginner\_tutorials/AddTwoInts.h"

[3](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_3)

[4](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_4) bool add(beginner\_tutorials::AddTwoInts::Request &req,

[5](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_5) beginner\_tutorials::AddTwoInts::Response &res)

[6](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_6) {

[7](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_7) res.sum = req.a + req.b;

[8](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_8) ROS\_INFO("request: x=%ld, y=%ld", (long int)req.a, (long int)req.b);

[9](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_9) ROS\_INFO("sending back response: [%ld]", (long int)res.sum);

[10](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_10) return **true**;

[11](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_11) }

[12](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_12)

[13](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_13) int main(int argc, char \*\*argv)

[14](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_14) {

[15](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_15) ros::init(argc, argv, "add\_two\_ints\_server");

[16](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_16) ros::NodeHandle n;

[17](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_17)

[18](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_18) ros::ServiceServer service = n.advertiseService("add\_two\_ints", add);

[19](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_19) ROS\_INFO("Ready to add two ints.");

[20](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_20) ros::spin();

[21](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_21)

[22](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_22) return 0;

[23](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-51b90df3945605c351da24a5ec712323d4d3dc5a_23) }

代码解释

现在，让我们来逐步分析代码。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29)

[1](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-44045d249f9a09eed4ca0e3c86767fcc080379f7_1) #include "ros/ros.h"

[2](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-44045d249f9a09eed4ca0e3c86767fcc080379f7_2) #include "beginner\_tutorials/AddTwoInts.h"

[3](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-44045d249f9a09eed4ca0e3c86767fcc080379f7_3)

beginner\_tutorials/AddTwoInts.h是由编译系统自动根据我们先前创建的srv文件生成的对应该srv文件的头文件。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29)

[4](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-c703b7a0cd73dc5e242b12211e5ef7ade318e6f7_4) bool add(beginner\_tutorials::AddTwoInts::Request &req,

[5](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-c703b7a0cd73dc5e242b12211e5ef7ade318e6f7_5) beginner\_tutorials::AddTwoInts::Response &res)

这个函数提供两个int值求和的服务，int值从request里面获取，而返回数据装入response内，这些数据类型都定义在srv文件内部，函数返回一个boolean值。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29)

[6](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-cab6c9ffa40cf9bd42fef4448b31f6feaa24464c_6) {

[7](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-cab6c9ffa40cf9bd42fef4448b31f6feaa24464c_7) res.sum = req.a + req.b;

[8](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-cab6c9ffa40cf9bd42fef4448b31f6feaa24464c_8) ROS\_INFO("request: x=%ld, y=%ld", (long int)req.a, (long int)req.b);

[9](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-cab6c9ffa40cf9bd42fef4448b31f6feaa24464c_9) ROS\_INFO("sending back response: [%ld]", (long int)res.sum);

[10](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-cab6c9ffa40cf9bd42fef4448b31f6feaa24464c_10) return **true**;

[11](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-cab6c9ffa40cf9bd42fef4448b31f6feaa24464c_11) }

现在，两个int值已经相加，并存入了response。然后一些关于request和response的信息被记录下来。最后，service完成计算后返回true值。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29)

[18](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-ac5d6b9af10384158d452506d30df9e87f3dbd88_18) ros::ServiceServer service = n.advertiseService("add\_two\_ints", add);

这里，service已经建立起来，并在ROS内发布出来。

编写Client节点

代码

在beginner\_tutorials包中创建src/add\_two\_ints\_client.cpp文件，并复制粘贴下面的代码：

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29)

[1](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_1) #include "ros/ros.h"

[2](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_2) #include "beginner\_tutorials/AddTwoInts.h"

[3](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_3) #include <cstdlib>

[4](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_4)

[5](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_5) int main(int argc, char \*\*argv)

[6](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_6) {

[7](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_7) ros::init(argc, argv, "add\_two\_ints\_client");

[8](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_8) if (argc != 3)

[9](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_9) {

[10](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_10) ROS\_INFO("usage: add\_two\_ints\_client X Y");

[11](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_11) return 1;

[12](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_12) }

[13](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_13)

[14](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_14) ros::NodeHandle n;

[15](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_15) ros::ServiceClient client = n.serviceClient<beginner\_tutorials::AddTwoInts>("add\_two\_ints");

[16](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_16) beginner\_tutorials::AddTwoInts srv;

[17](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_17) srv.request.a = atoll(argv[1]);

[18](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_18) srv.request.b = atoll(argv[2]);

[19](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_19) if (client.call(srv))

[20](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_20) {

[21](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_21) ROS\_INFO("Sum: %ld", (long int)srv.response.sum);

[22](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_22) }

[23](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_23) else

[24](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_24) {

[25](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_25) ROS\_ERROR("Failed to call service add\_two\_ints");

[26](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_26) return 1;

[27](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_27) }

[28](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_28)

[29](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_29) return 0;

[30](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-f2a39abe84f67d951ff455fdbeda1ad12e9783e4_30) }

代码解释

现在，让我们来逐步分析代码。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29)

[15](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-ae5c23e9275f52f11e35b1c440a1b28eb4e05c6b_15) ros::ServiceClient client = n.serviceClient<beginner\_tutorials::AddTwoInts>("add\_two\_ints");

这段代码为add\_two\_ints service创建一个client。ros::ServiceClient 对象待会用来调用service。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29)

[16](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-6c61651582a4391fbbdc9c59b97b6557ba35c65a_16) beginner\_tutorials::AddTwoInts srv;

[17](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-6c61651582a4391fbbdc9c59b97b6557ba35c65a_17) srv.request.a = atoll(argv[1]);

[18](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-6c61651582a4391fbbdc9c59b97b6557ba35c65a_18) srv.request.b = atoll(argv[2]);

这里，我们实例化一个由ROS编译系统自动生成的service类，并给其request成员赋值。一个service类包含两个成员request和response。同时也包括两个类定义Request和Response。

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29)

[19](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29#cn.2BAC8-roscpp_tutorials.2BAC8-Tutorials.2BAC8-WritingServiceClient.CA-449923a1ec1e665c95b5e8fe0e72050ab09d0f7a_19) if (client.call(srv))

这段代码是在调用service。由于service的调用是模态过程（调用的时候占用进程阻止其他代码的执行），一旦调用完成，将返回调用结果。如果service调用成功，call()函数将返回true，srv.response里面的值将是合法的值。如果调用失败，call()函数将返回false，srv.response里面的值将是非法的。

编译节点

再来编辑一下beginner\_tutorials里面的CMakeLists.txt：

$ rosed beginner\_tutorials CMakeLists.txt

将下面的内容加载文件末尾：

rosbuild\_add\_executable(add\_two\_ints\_server src/add\_two\_ints\_server.cpp)

rosbuild\_add\_executable(add\_two\_ints\_client src/add\_two\_ints\_client.cpp)

这段代码将生成两个可执行程序"add\_two\_ints\_server"和"add\_two\_ints\_client"，这两个可执行程序默认被放在"bin"文件夹下。

关于在ROS上面使用CMake的更多信息，可以参考[CMakeLists](http://wiki.ros.org/CMakeLists) 现在，运行make命令：

$ make

编译节点

再来编辑一下beginner\_tutorials里面的CMakeLists.txt，文件位于~/catkin\_ws/src/beginner\_tutorials/CMakeLists.txt，并将下面的代码添加在文件末尾：

[*https://raw.github.com/ros/catkin\_tutorials/master/create\_package\_srvclient/catkin\_ws/src/beginner\_tutorials/CMakeLists.txt*](https://raw.github.com/ros/catkin_tutorials/master/create_package_srvclient/catkin_ws/src/beginner_tutorials/CMakeLists.txt)

[Toggle line numbers](http://wiki.ros.org/cn/ROS/Tutorials/WritingServiceClient%28c%2B%2B%29)

add\_executable(add\_two\_ints\_server src/add\_two\_ints\_server.cpp)

target\_link\_libraries(add\_two\_ints\_server ${catkin\_LIBRARIES})

add\_dependencies(add\_two\_ints\_server beginner\_tutorials\_gencpp)

add\_executable(add\_two\_ints\_client src/add\_two\_ints\_client.cpp)

target\_link\_libraries(add\_two\_ints\_client ${catkin\_LIBRARIES})

add\_dependencies(add\_two\_ints\_client beginner\_tutorials\_gencpp)

这段代码将生成两个可执行程序"add\_two\_ints\_server"和"add\_two\_ints\_client"，这两个可执行程序默认被放在你的[devel space](http://wiki.ros.org/catkin/workspaces#Development_.28Devel.29_Space)下的包目录下，默认为~/catkin\_ws/devel/lib/share/<package name>。你可以直接调用可执行程序，或者使用rosrun命令去调用它们。它们不会被装在<prefix>/bin目录下，因为当你在你的系统里安装这个包的时候，这样做会污染PATH变量。如果你希望在安装的时候你的可执行程序在PATH变量里面，你需要设置一下install target，请参考：[catkin/CMakeLists.txt](http://wiki.ros.org/catkin/CMakeLists.txt)

关于[CMakeLists.txt](http://wiki.ros.org/catkin/CMakeLists.txt)文件更详细的描述请参考：[catkin/CMakeLists.txt](http://wiki.ros.org/catkin/CMakeLists.txt)

现在运行catkin\_make命令：

# In your catkin workspace

cd ~/catkin\_ws

catkin\_make

# 测试服务器和客户端

## **Running the Service**

Let's start by running the service:

$ rosrun beginner\_tutorials add\_two\_ints\_server (C++)

$ rosrun beginner\_tutorials add\_two\_ints\_server.py (Python)

You should see something similar to:

Ready to add two ints.

## **Running the Client**

Now let's run the client with the necessary arguments:

$ rosrun beginner\_tutorials add\_two\_ints\_client 1 3 (C++)

$ rosrun beginner\_tutorials add\_two\_ints\_client.py 1 3 (Python)

You should see something similar to:

Requesting 1+3

1 + 3 = 4

Now that you've successfully run your first server and client, let's learn how to [record and play back data](http://wiki.ros.org/ROS/Tutorials/Recording%20and%20playing%20back%20data).

# 获得帮助

我们已经接触到不少的ROS工具了。有时候很难记住他们所需要的参数。还好大多数ROS工具都提供了帮助。

输入:

$ rosmsg -h

* 你可以看到一系列的rosmsg子命令.
* Commands:
* rosmsg show Show message description
* rosmsg users Find files that use message
* rosmsg md5 Display message md5sum
* rosmsg package List messages in a package

rosmsg packages List packages that contain messages

同样你也可以获得子命令的帮助：

$ rosmsg show -h

* 这会现实rosmsg show 所需的参数：
* Usage: rosmsg show [options] <message type>
* Options:
* -h, --help show this help message and exit
* -r, --raw show raw message text, including comments

# 网上的文章：

## 1.环境变量，安装等

1.ROS安装部分：

大部分问题集中于apt-get 使用上

首先看下这个：<http://jingyan.baidu.com/article/22a299b51648e09e19376ae7.html>

遇到类似 unmet dependency 时候，说明软件包安装的依赖出问题了，估计是误删或者ROS安装的时候要求软件版本有误，这个主意一下：

[](http://blog.exbot.net/wp-content/uploads/2015/06/%E5%9B%BE%E7%89%871.png)

主要集中于ubuntu 14.04 14.04.1 与之后的 ubuntu 14.04.2 在opengl版本上的一些分歧，我也被坑过的.......

遇到hash sum mismatch 的时候，有可能是你apt-get update 中途退出，直接装的时候会出这个问题：<http://askubuntu.com/questions/41605/trouble-downloading-packages-list-due-to-a-hash-sum-mismatch-error>

总之，apt-get出问题了，都不能算是ROS的问题，问ROS社区帮助不是很大偶......google是大家的好基友。

关于ROS deb软件包的概念：ROS 是一堆耦合程度很松的软件集合（除了一些通讯核心库和基础库，很多功能性的软件包之间完全独立）。安装位置一般在/opt/ros 下。ROS deb 安装是没有源码的，寻找源码要到wiki上的 source：标签寻找（详见之后的如何使用ROS wiki）

* ROS基础使用部分：

首先处理环境变量的问题，一开始，按照ROS install 做完之后，试着敲一下ROS的命令行工具来检测初始环境变量是否完好，比如roscore。

一般来说，使用

export | grep ROS

[](http://blog.exbot.net/wp-content/uploads/2015/06/%E5%9B%BE%E7%89%872.png)[图片3](http://blog.exbot.net/wp-content/uploads/2015/06/%E5%9B%BE%E7%89%873.png)[图片4](http://blog.exbot.net/wp-content/uploads/2015/06/%E5%9B%BE%E7%89%874.png)

（这是我没配置进阶包类似于rosjava ros-android 没有设置多机等等的基础ROS环境设置，新手们可以照着这个看自己哪个地方有问题）

而缺哪些环境变量会出现哪种问题，这个实在是千奇百怪，我细说一下各个环境变量的含义，大家也不要太浮躁，耐心看完自己处理问题最好。

与使用ROS有关的环境变量：

运行时用到的：

$PATH ：这个是你命令行中命令查找路径，换句话说，如果PATH 中没有你ROS相关的目录的话，是没办法使用命令行工具的，一般roscore not found 就是这个有问题。

$PYTHONPATH ：这个是python寻找package时的路径，如果有误的话，import ROS相关的包就会找不到

构建时用到的：

$CPATH  $CMAKE\_PREFIX\_PATH  $LD\_LIBRARY\_PATH  $PKG\_CONFIG\_PATH

$CMAKE\_PREFIX\_PATH ：一堆主要和cmake有关的东西，出错也基本是一起出，出问题了自己的catkin\_ws 构建就会出各种错。

ROS本身的环境变量：

1》ROS文件系统相关的：

$ROS\_PACKAGE\_PATH：（基本这就是50%以上问题的元凶）这个环境变量主要影响ROS文件系统工具查找ros package，指定几个地址去找相应的package，一般自建的catkin\_ws 不把它的路径加入的话，是无法通过ROS命令行工具以及roslaunch 来运行或者调用里面的包的。（简单来说就是各种not found）

2》ROS核心通讯相关：

$ROS\_ROOT：给出ROS 核心通讯组件的路径

$ROS\_MASTER\_URI：给出ROS MASTER的URI，一般是本机上，11311端口。

3》ROS多机通讯相关：

$ROS\_IP/$ROS\_HOSTNAME: 一般选一个用，多机连一起的时候，设置好这个就可以了，用于标示自己的在网络中的位置。

$ROS\_NAMESPACE:表示本机上的namespace，本机上node中的name都会有这个namespace做前缀。

4》工具相关：

$ROS\_WORKSPACE：使用rosinstall 和 rosws工具的同学看好了，这个表示这些工具的作用区域，而使用rosws init 会设置这个变量，如果没有设置这个以上工具使用会有问题的。

（还有一些新手基本不会用到的高阶环境变量就不说了。。。。）

仔细看$ROS\_PACKAGE\_PATH

然后就要说道这些环境变量的设置了：

一开始在apt-get install ros-desktop-full之后，教程就会让你

source /opt/ros/<distro>/setup.bash （大家一定要懂一点linux，不懂的我解释一下，就是执行setup.bash这个脚本）

这个脚本的作用：（翻译自脚本注释）设置各种环境变量，设置工具链（主要就是之前cmake一堆环境变量和设置），**它会尽可能的撤销已有的设置的改变。（先source 自己工作空间的setup.bash 再source  opt安装的setup.bash 关于自己工作空间的一些设置，比如ROS\_PACKAGE\_PATH就会没有的）**

然后使用catkin\_init\_workspace ， catkin\_make 后会有source /home/<user>/catkin\_ws/devel/setup.bash

之前的设置（opt那边的ROS setup）会overlay 到home这边的 setup里，所以基本source了自己工作空间里的setup.bash，以上说的环境变量以及设置都会做好，再有那些特殊需求，自己export 设置就好了。

然后就是关于/hom/<user>/文件夹下.bashrc是个啥：

这个文件是关于这个用户的bash shell 设置，每次以当前用户打开命令行都会运行一遍这个脚本，大家为了不用每次都source一遍，所以把命令写到了这个脚本里面。

关于这个脚本一些更具体的知识：<http://blog.163.com/wang_hai_fei/blog/static/309020312008728333912/>

（对新手的忠告：一般出问题90%都跟以上这些设置有关，先检查这些，自己体会做做笔记，更加重要的是要先了解linux和一些周边的知识，学习ROS不需要很着急，着急反而事倍功半）

* ROS 里进阶使用：

Ros主要工作流程 （坑略大）

Urdf的概念（这个得自己看机器人学导论然后再说才说的清楚）

Roslaunch 注意事项（这个坑略大，以后再填）

* gazebo使用问题：

最大的问题对于新手来说，就是gazebo开始的时候会去连接gazebo官网的远端模型数据库（remote model database），而国内你懂的，基本连不上，所以gazebo就会自己崩掉（gazebo 4以上版本不会出现了）.（症状一般是wait timeout一类的提示，然后没有出来仿真界面自己退出了）

这个解决方法是：

先下载gazebo的模型文件集合，然后添加环境变量

[图片5](http://blog.exbot.net/wp-content/uploads/2015/06/%E5%9B%BE%E7%89%875.png)

完美解决。至于之后的各种问题，再讨论吧。。。。

* 怎么看wiki：

（终于到了本文的重点）

wiki的结构是这样的：基本按照package为单元来介绍

举个例子：

[](http://blog.exbot.net/wp-content/uploads/2015/06/%E5%9B%BE%E7%89%876.png)

找源码看 source ，找单独的package教程看右边的package link

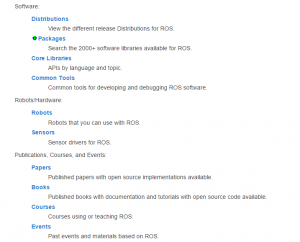
有拿oxygen生成的api文档，相关的教程（如果有的话），也有一些比较重要的信息，这些在源码文件中都找得到。。。。

下面格式基本是（1）算法简介，（2）使用方法，（3）节点信息（订阅发布那些信息，提供哪些service，有哪些参数）

查找具体功能包一般看

[图片7](http://blog.exbot.net/wp-content/uploads/2015/06/%E5%9B%BE%E7%89%877.png)

Browse software ,里面有各个版本发型的所有package列表，可以通过这个直接进入你感兴趣的package文档。

[](http://blog.exbot.net/wp-content/uploads/2015/06/%E5%9B%BE%E7%89%878.png)[](http://blog.exbot.net/wp-content/uploads/2015/06/%E5%9B%BE%E7%89%879.png)

查找ROS基础库文档，也可以通过之前的方式，但是找到略麻烦

APIs页面的编排还是非常具有逻辑性的，这样也可以快速定位到你需要的基础库。

Wiki主页，找基础库看core library ，找具体硬件支持看robots，具体传感器支持找sensors（虽然支持版本有些旧）最后都是定位到具体的package上，具体参照package的阅读方法。

比较难找的但是常用的：

Roslaunch xml文件标签说明：<http://wiki.ros.org/roslaunch/XML>

Urdf xml 文件标签说明：<http://wiki.ros.org/urdf/XML>

Roscpp api 文档：<http://docs.ros.org/jade/api/roscpp/html/>

Rospy api 文档：<http://docs.ros.org/jade/api/rospy/html/>

## 2环境变量ROS\_PACKAGE\_PATH

昨天刚成功安装了ardrone\_autonomy 和 tum\_ardrone，运行也是通过了。今天又尝试了一下昨天的命令，结果发现tum\_ardrone居然又运行不了了，郁闷！

说是没有在环境变量$ROS\_PACKAGE\_PATH中所有的路径中都找不到tum\_ardrone这个package，然后就想到，是否应该是要把保存tum\_ardrone的路径也添加到环境变量中呢？重新修改了～/.bashrc中的export $ROS\_PACKAGE\_PATH的内容，添加了tum\_ardrone的目录～/.catkin\_ws/src，然后再次运行命令，终于再次成功了。

通过这个问题也认识了$ROS\_PACKAGE\_PATH的重要性，对于需要全局引用的文件如包等的路径，都应当要添加到环境变量中去。虽然很短，也还是记录下来吧！加油。