

创建 ROS 程序包

AI 航 团队

1. 一个 catkin 程序包的组成

一个程序包要想称为 catkin 程序包必须符合以下要求：

1) 该程序包必须包含 catkin compliant package.xml 文件。这个 package.xml 文件提供有关程序包的元信息。

2) 程序包必须包含一个 catkin 版本的 CMakeLists.txt 文件，而 Catkin metapackages 中必须包含一个对 CMakeList.txt 文件的引用。

3) 每个目录下只能有一个程序包。这意味着在同一个目录下不能有嵌套的或者多个程序包存在。最简单的程序包也许看起来就像这样：

```
my_package/  
    CMakeLists.txt  
    package.xml
```

2. 一个在 catkin 工作空间中的程序

开发 catkin 程序包的一个推荐方法是使用 catkin 工作空间，但是你也可以单独开发 (standalone)catkin 软件包。一个简单的工作空间也许看起来像这样：

在继续本教程之前请先按照创建 catkin 工作空间教程创建一个空白的 catkin 工作空间。

```
workspace_folder/    -- WORKSPACE
  src/               -- SOURCE SPACE
  CMakeLists.txt     -- 'Toplevel' CMake file, provided by catkin
package_1/           CMakeLists.txt -- CMakeLists.txt file for package_1
  package.xml        -- Package manifest for package_1
...
package_n/
  CMakeLists.txt     -- CMakeLists.txt file for package_n
    package.xml      -- Package manifest for package_n
```

3. 创建一个 catkin 程序包

本部分教程将演示如何使用 `catkin_create_pkg` 命令来创建一个新的 catkin 程序包以及创建之后都能做些什么。

首先切换到之前通过创建 catkin 工作空间教程创建的 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
```

示例：

```
yt@yt-UNO-2483G-453AE:~/catkin_ws/src/beginner_tutorials$ cd ~/catkin_ws/  
yt@yt-UNO-2483G-453AE:~/catkin_ws$ cd ~/catkin_ws/src  
yt@yt-UNO-2483G-453AE:~/catkin_ws/src$ catkin_create_pkg beginner_tutorials1 std_msgs rospy roscpp  
Created file beginner_tutorials1/CMakeLists.txt  
Created file beginner_tutorials1/package.xml  
Created folder beginner_tutorials1/include/beginner_tutorials1  
Created folder beginner_tutorials1/src  
Successfully created files in /home/yt/catkin_ws/src/beginner_tutorials1. Please  
adjust the values in package.xml.  
yt@yt-UNO-2483G-453AE:~/catkin_ws/src$
```

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` 文件和一个 `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` 中有描述。

4. 程序包依赖关系

4.1 一级依赖

现在我们可以使用 `rospack` 命令工具来查看一级依赖包。

```
$ rospack depends1 beginner_tutorials
```

就像你看到的，`rospack` 列出了在运行 `catkin_create_pkg` 命令时作为参数的依赖包，这些依赖包随后保存在 `package.xml` 文件中。

```
std_msgs  
rospy  
roscpp
```

```
$ roscd beginner_tutorials
```

```
$ cat package.xml
```

```
<package>
...
  <buildtool_depend>catkin</buildtool_depend> <build_depend>roscpp
</build_depend>
<build_depend>rospy</build_depend>
  <build_depend>std_msgs</build_depend>
...
</package>
```

4.2 间接依赖

在很多情况中，一个依赖包还会有它自己的依赖包，比如，**rospy** 还有其它依赖包。

```
$ rospack depends1 rospy
```

```
genpy
rosgraph
rosgraph_msgs
roslib
std_msgs
```

一个程序包还可以有好几个间接的依赖包，幸运的是使用 **rospack** 可以递归检测出所有的依赖包。

```
$ rospack depends beginner_tutorials
cpp_common
rostime
roscpp_traits
roscpp_serialization
genmsg
genpy
message_runtime
roscconsole
std_msgsrosgaph_msgs
xmlrpcpp
roscpp
rosgraph
catkin
rospack
roslib
```

5. 自定义你的程序包

本部分教程将剖析 **catkin_create_pkg** 命令生成的每个文件并详细描述这些文件的组成

部分以及如何自定义这些文件。

5.1 自定义 package.xml

自动生成的 package.xml 文件应该在你的新程序包中。现在让我们一起来看看新生成的 package.xml 文件以及每一个需要你注意的标签元素。

```
yt@yt-UN0-2483G-453AE:~/catkin_ws/src/beginner_tutorials$ cd ~/catkin_ws/  
yt@yt-UN0-2483G-453AE:~/catkin_ws$ cd ~/catkin_ws/src  
yt@yt-UN0-2483G-453AE:~/catkin_ws/src$ catkin_create_pkg beginner_tutorials1 std  
_msgs rospy roscpp  
Created file beginner_tutorials1/CMakeLists.txt  
Created file beginner_tutorials1/package.xml  
Created folder beginner_tutorials1/include/beginner_tutorials1  
Created folder beginner_tutorials1/src  
Successfully created files in /home/yt/catkin_ws/src/beginner_tutorials1. Please  
adjust the values in package.xml.  
yt@yt-UN0-2483G-453AE:~/catkin_ws/src$
```

5.1.1 描述标签

首先更新描述标签

```
<description>The beginner_tutorials package</description>
```

将描述信息修改为任何你喜欢的内容，但是按照约定第一句话应该简短一些，因为它覆盖了程序包的范围。如果用一句话难以描述完全那就需要换行了。

5.1.2 维护者标签

接下来是维护者标签

```
7 <!-- One maintainer tag required, multiple allowed, one person per tag -->  
8 <!-- Example: -->  
9 <!-- <maintainer email="jane.doe@example.com">Jane Doe</maintainer> -->  
10 <maintainer email="user@todo.todo">user</maintainer>
```

这是 package.xml 中要求填写的一个重要标签，因为它能够让其他人联系到程序包的相关人员。至少需要填写一个维护者名称，但如果有需要的话你可以添加多个。除了在标签里面填写维护者的名称外，还应该在标签的 email 属性中填写邮箱地址：

```
7 <maintainer email="you@yourdomain.tld">Your Name</maintainer>
```

5.1.3 许可标签

再接下来是许可标签，同样的也需要：

切换行号显示

```
12 <!-- One license tag required, multiple allowed, one license per tag -->
13 <!-- Commonly used license strings: -->
14 <!-- BSD, MIT, Boost Software License, GPLv2, GPLv3, LGPLv2.1, LGPLv3 -->
15 <license>TODO</license>
```

你应该选择一种许可协议并将它填写到这里。一些常见的开源许可协议有 BSD、MIT、Boost Software License、GPLv2、GPLv3、LGPLv2.1 和 LGPLv3。你可以在 Open Source Initiative 中阅读其中的若干个许可协议的相关信息。对于本教程我们将使用 BSD 协议，因为 ROS 核心组件的剩余部分已经使用了该协议：

切换行号显示

```
8 <license>BSD</license>
```

5.1.4 依赖项标签

接下来的标签用来描述程序包的各种依赖项，这些依赖项分为 `build_depend`、`buildtool_depend`、`run_depend`、`test_depend`。关于这些标签的更详细介绍请参考 Catkin Dependencies 相关的文档。在之前的操作中，因为我们将 `std_msgs`、`roscpp`、和 `rospy` 作为 `catkin_create_pkg` 命令的参数，所以生成的依赖项看起来如下：

```
27 <!-- The *_depend tags are used to specify dependencies -->
28 <!-- Dependencies can be catkin packages or system dependencies -->
29 <!-- Examples: -->
30 <!-- Use build_depend for packages you need at compile time: -->
31 <!-- <build_depend>genmsg</build_depend> -->
32 <!-- Use buildtool_depend for build tool packages: -->
33 <!-- <buildtool_depend>catkin</buildtool_depend> -->
34 <!-- Use exec_depend for packages you need at runtime: -->
35 <!-- <exec_depend>python-yaml</exec_depend> -->
36 <!-- Use test_depend for packages you need only for testing: -->
37 <!-- <test_depend>gtest</test_depend> -->
38 <buildtool_depend>catkin</buildtool_depend>
39 <build_depend>roscpp</build_depend>
40 <build_depend>rospy</build_depend>
41 <build_depend>std_msgs</build_depend>
```

除了 `catkin` 中默认提供的 `buildtool_depend`，所有我们列出的依赖包都已经被添加到 `build_depend` 标签中。在本例中，因为在编译和运行时我们需要用到所有指定的依赖包，

因此还需要将每一个依赖包分别添加到 `run_depend` 标签中：

```

切换行号显示
12 <buildtool_depend>catkin</buildtool_depend>
13
14 <build_depend>roscpp</build_depend>
15 <build_depend>rospy</build_depend>
16 <build_depend>std_msgs</build_depend>
17
18 <exec_depend>roscpp</exec_depend>
19 <exec_depend>rospy</exec_depend>
20 <exec_depend>std_msgs</exec_depend>

```

5.1.5 最后完成的 package.xml

现在看下面最后去掉了注释和未使用标签后的 `package.xml` 文件就显得更加简洁了：

```

切换行号显示
1 <?xml version="1.0"?>
2 <package format="2">
3   <name>beginner_tutorials</name>
4   <version>0.1.0</version>
5   <description>The beginner_tutorials package</description>
6
7   <maintainer email="you@yourdomain.tld">Your Name</maintainer>
8   <license>BSD</license>
9   <url type="website">http://wiki.ros.org/beginner_tutorials</url>
10  <author email="you@yourdomain.tld">Jane Doe</author>
11
12  <buildtool_depend>catkin</buildtool_depend>
13
14  <build_depend>roscpp</build_depend>
15  <build_depend>rospy</build_depend>
16  <build_depend>std_msgs</build_depend>
17
18  <exec_depend>roscpp</exec_depend>
19  <exec_depend>rospy</exec_depend>
20  <exec_depend>std_msgs</exec_depend>
21
22 </package>

```

5.2 自定义 CMakeList.txt

到此，这个包含程序包元信息的 `package.xml` 文件已经按照需要完成了裁剪整理，现在

你可以继续下面的教程了。`catkin_create_pkg` 命令生成的 `CMakeLists.txt` 文件将在后续关于编译 ROS 程序代码的教程中讲述。