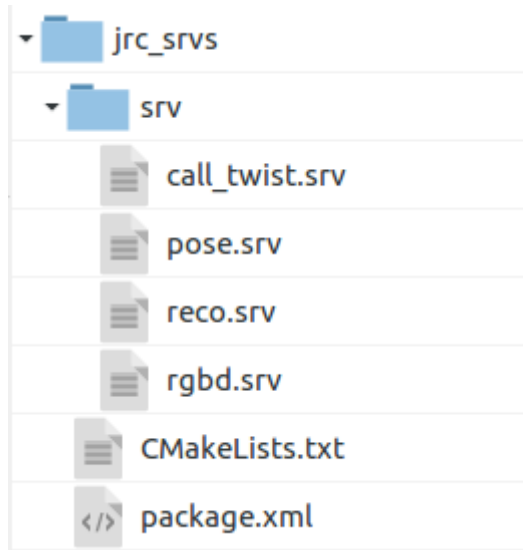


规范

Node 统一命名

首先所有的 `.srv` 文件 都放在 `/jrc_srvs` 下



并在CMakeLists.txt 中注册自己的srv文件：

```
add_service_files(
  FILES
  ### 所有的 .srv 文件
  reco.srv
  pose.srv
  rgbd.srv
  call_twist.srv
)
```

这样我们所有的自定义srv文件都以 `jrc_srvs` 引用：

```
#include "jrc_srvs/call_twist.h"
#include "jrc_srvs/call_grasp.h"
```

所有的 提供 `service` 的 `package` 命名都叫 `XXX_server`
主程序的源文件命名叫 `XXX_server.cpp` ,
`node` 叫 `XXX_server_node`

```
add_executable(ur_ee_server_node src/ur_ee_server.cpp)

target_link_libraries(ur_ee_service_node
  ${catkin_LIBRARIES}
)
```

如何解决之前大家遇到的“要先编srv包，再编node包”这个麻烦的问题：

其实很好解决。你在自己的包内解决好了也为其他人编译提供了便利。
按照我给的步骤来，以我的 `jrc_main` 为例：

划重点！！！！

1. 改 CMakeList.txt

在 `find_package` 处添加我们管理所有.srv文件的包名：`jrc_rvs`

```
find_package(catkin REQUIRED COMPONENTS
  image_transport
  cv_bridge
  roscpp
  rospy
  sensor_msgs
  std_msgs
  message_generation
  tf
  ### 所有自定义的 srv 应该在这声明：
  jrc_srvs
)
```

在 `add_executable` 与 `target_link_libraries` 中间加上这句话：

```
add_executable(jrc_main_node
  src/jrc_main.cpp
  src/AGV.cpp
)

add_dependencies(jrc_main_node jrc_srvs_gencpp)

target_link_libraries(jrc_main_node
  ${catkin_LIBRARIES}
  ${OpenCV_LIBS}
  ${PCL_LIBRARIES}
)
```

2. 改 package.xml

在 `buildtool_depend` 最后加上我们的包名 `jrc_rvs`

```
<!-- Use doc_depend for packages you need only for building docs -->
<!--   <doc_depend>doxygen</doc_depend> -->
<buildtool_depend>catkin</buildtool_depend>

<build_depend>roscpp</build_depend>
<build_depend>rospy</build_depend>
<build_depend>sensor_msgs</build_depend>
<build_depend>std_msgs</build_depend>
<build_depend>message_generation</build_depend>
<build_depend>jrc_srvs</build_depend>
```

在 `exec_depend` 最后加上我们的包名 `jrc_rvs`

```
<exec_depend>roscpp</exec_depend>
<exec_depend>rospy</exec_depend>
<exec_depend>sensor_msgs</exec_depend>
<exec_depend>std_msgs</exec_depend>
<exec_depend>message_runtime</exec_depend>
<exec_depend>jrc_srvs</exec_depend>
```

CMakeList.txt 与 package.xml 版本统

其次，所有的 CMakeLists.txt 前面保证这两句：

```
cmake_minimum_required(VERSION 2.8)
add_compile_options(-std=c++11)
```

所有的 Package: 保证“

```
<?xml version="1.0"?>
<package format="2">
<name>pose_server</name>
<version>0.0.0</version>
<description>The pose_server package</description>
```