

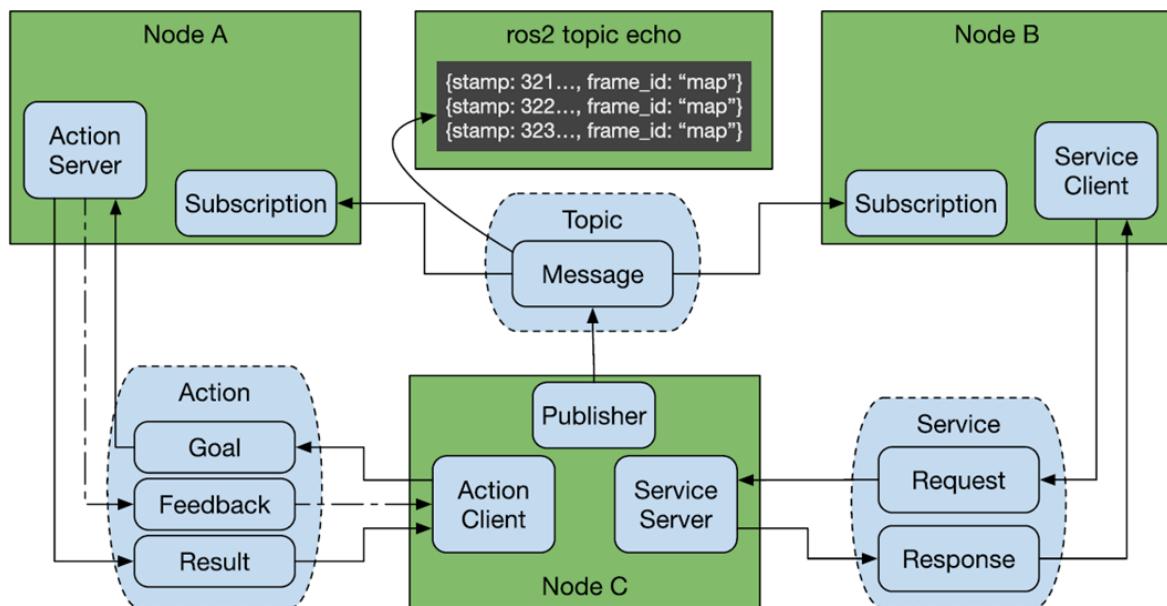
# 10、ROS2自定义接口消息

## 1、通讯接口简介

在ROS系统中，无论话题、服务还是动作，都会用到一个重要的概念——通信接口。

通信并不是一个人自言自语，而是两个甚至更多个人，你来我往的交流，交流的内容是什么呢？为了让大家更好理解，我们可以给传递的数据定义一个标准的结构，这就是通信接口。

接口可以让程序之间的依赖降低，便于我们使用别人的代码，也方便别人使用我们的代码，这就是ROS的核心目标，减少重复造轮子。



ROS有三种常用的通信机制，分别是话题、服务、动作，通过每一种通信种定义的接口，各种节点才能有机的联系到一起。

## 2、创建自定义接口流程

自定义接口消息的流程与在功能包中编写可执行程序的流程类似，主要步骤如下：

1. 创建接口功能包；
2. 创建并编辑 .msg文件、.srv文件、.action文件
3. 编辑配置文件；
4. 编译；
5. 测试。

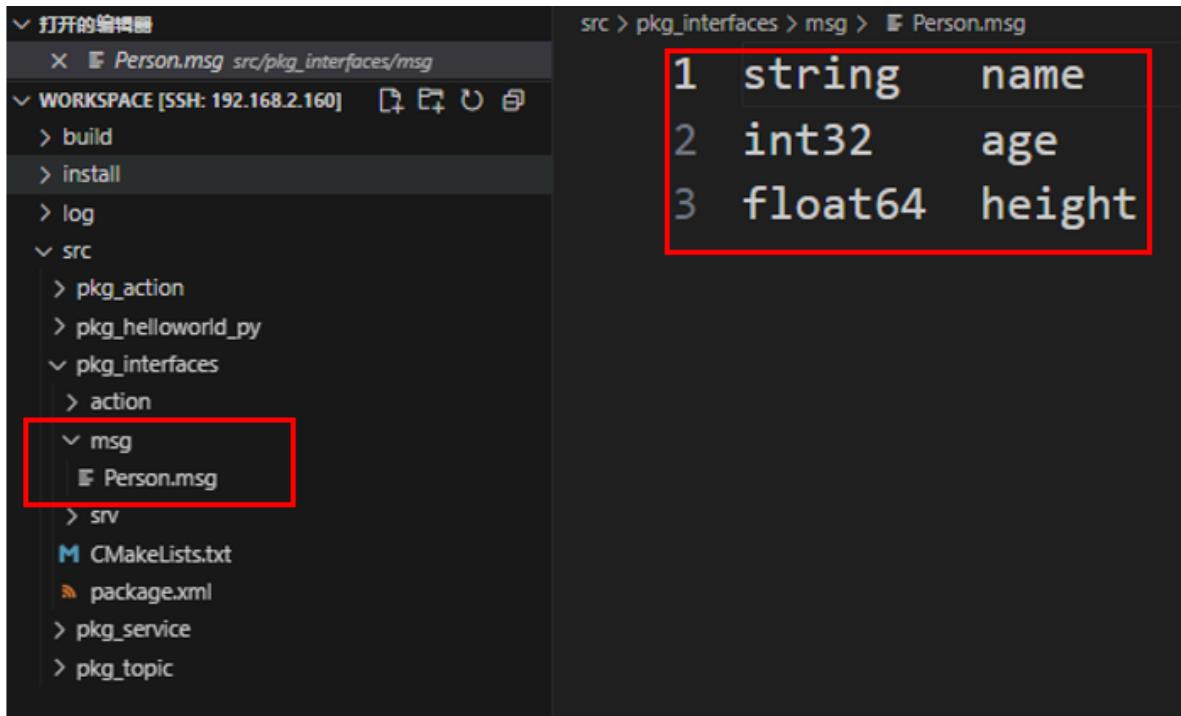
## 3、创建动作通信自定义接口

在【11、ROS2动作通讯服务端实现】案例中，我们已经演示过如果创建动作通讯接口的完整流程，大家可以先回去复习一下，这里就不再赘述。

## 4、创建话题通信自定义接口

1、在【ROS2动作通讯服务端实现】课程中我们已经创建了自定义接口功能包，功能包pkg\_interfaces下新建msg文件夹，msg文件夹下新建**Person.msg**文件，文件中输入如下内容：

```
string name
int32 age
float64 height
```



2、在package.xml和CMakeLists.txt中添加如下配置：

- CMakeLists.txt

```
rosidl_generate_interfaces(${PROJECT_NAME}
    "action/Progress.action"
    "msg/Person.msg"
)
```

- package.xml

```
<buildtool_depend>rosidl_default_generators</buildtool_depend>
<exec_depend>rosidl_default_runtime</exec_depend>
<depend>action_msgs</depend>
<member_of_group>rosidl_interface_packages</member_of_group>
```

```

M CMakeLists.txt x ...
yahboomcar.ws > src > pkg_interfaces > M CMakeLists.txt
10 if(NOT CMAKE_CXX_STANDARD)
11 | set(CMAKE_CXX_STANDARD 14)
12 endif()
13
14 if(CMAKE_COMPILER_IS_GNUCXX OR CMAKE_CXX_COMPILER_ID MATCHES "Clang")
15 | add_compile_options(-Wall -Wextra -Wpedantic)
16 endif()
17
18 # find dependencies
19 find_package(ament_cmake REQUIRED)
20 find_package(roslidl_default_generators REQUIRED)
21 # uncomment the following section in order to fill in
22 # further dependencies manually.
23 # find_package(<dependency> REQUIRED)
24
25 rosidl_generate_interfaces(${PROJECT_NAME})
26 "action/Progress.action"
27 "msg/Person.msg"
28 )
29
30 if(BUILD_TESTING)
31 find_package(ament_lint_auto REQUIRED)
32 # the following line skips the linter which checks for copyrights
33 # uncomment the line when a copyright and license is not present in all source

```

package.xml x ...

```

yahboomcar.ws > src > pkg_interfaces > package.xml
1 <?xml version="1.0"?>
2 <xmldoc-model href="http://download.ros.org/schema/package_format3.xsd" schematype="external">
3 <package format="3">
4   <name>pkg_interfaces</name>
5   <version>0.0.0</version>
6   <description>TODO: Package description</description>
7   <maintainer email="1461190907@qq.com">root</maintainer>
8   <license>TODO: License declaration</license>
9
10  <buildtool_depend>ament_cmake</buildtool_depend>
11
12  <test_depend>ament_lint_auto</test_depend>
13  <test_depend>ament_lint_common</test_depend>
14
15  <buildtool_depend>roslidl_default_generators</buildtool_depend>
16  <exec_depend>roslidl_default_runtime</exec_depend>
17  <depend>action_msgs</depend>
18  <member_of_group>roslidl_interface_packages</member_of_group>
19
20  <export>
21    <build_type>ament_cmake</build_type>
22  </export>
23 </package>

```

### 3、终端中进入当前工作空间，编译功能包：

```
colcon build --packages-select pkg_interfaces
```

```

`colcon build` successful
/home/yahboom/workspace

yahboom@yahboom-virtual-machine: ~/workspace
yahboom@yahboom-virtual-machine: ~/workspace 94x24
yahboom@yahboom-virtual-machine:~/workspace$ colcon build --packages-select pkg_interfaces
[0.252s] WARNING:colcon.colcon_core.package_selection:Some selected packages are already built
in one or more underlay workspaces:
'pkg_interfaces' is in: /home/yahboom/workspace/install/pkg_interfaces
If a package in a merged underlay workspace is overridden and it installs headers, then all pa
ckages in the overlay must sort their include directories by workspace order. Failure to do so
may result in build failures or undefined behavior at run time.
If the overridden package is used by another package in any underlay, then the overriding pack
age in the overlay must be API and ABI compatible or undefined behavior at run time may occur.

If you understand the risks and want to override a package anyways, add the following to the c
ommand line:
--allow-override pkg_interfaces

This may be promoted to an error in a future release of colcon-override-check.
Starting >>> pkg_interfaces
Finished <<< pkg_interfaces [0.32s]

Summary: 1 package finished [0.52s]

```

### 4、测试接口是否正常

- 先刷新环境变量

```
source install/setup.bash
```

- 查看接口类型

```
ros2 interface show pkg_interfaces/msg/Person
```

正常情况下，终端将会输出与 Person.msg 文件一致的内容。

```

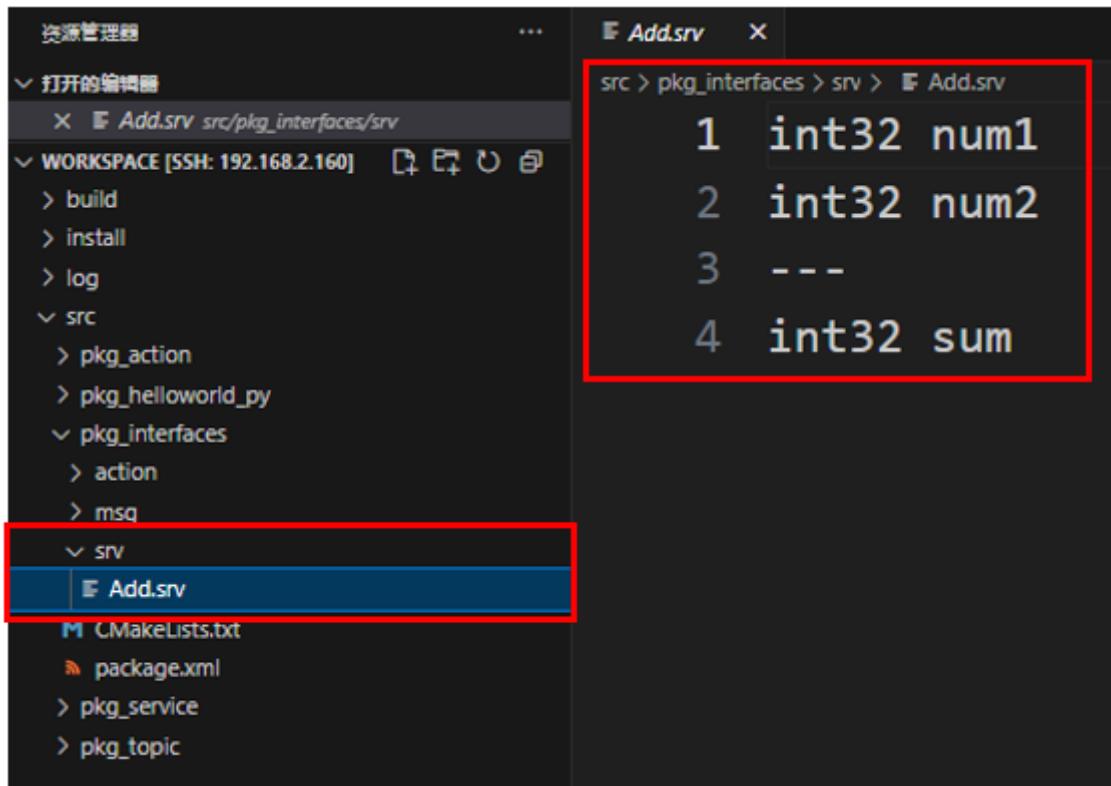
root@unbutu:~/yahboomcar_ros2_ws/yahboomcar_ws# ros2 interface show pkg_interfaces/msg/Person
string name
int32 age
float64 height
root@unbutu:~/yahboomcar_ros2_ws/yahboomcar_ws#

```

## 5、创建服务通信自定义接口

1、在【ROS2动作通讯服务端实现】课程中我们已经创建了自定义接口功能包，功能包pkg\_interfaces下新建srv文件夹，srv文件夹下新建Add.srv文件，文件中输入如下内容：

```
int32 num1
int32 num2
---
int32 sum
```



2、在package.xml和CMakeLists.txt中添加如下配置：

- CMakeLists.txt

```
rosidl_generate_interfaces(${PROJECT_NAME}
    "action/Progress.action"
    "msg/Person.msg"
    "srv/Add.srv"
)
```

- package.xml

```
<buildtool_depend>rosidl_default_generators</buildtool_depend>
<exec_depend>rosidl_default_runtime</exec_depend>
<depend>action_msgs</depend>
<member_of_group>rosidl_interface_packages</member_of_group>
```

```

M CMakeLists.txt x
yahboomcar_ws > src > pkg_interfaces > M CMakeLists.txt
15 | add_compile_options(-Wall -Wextra -Wpedantic)
16 endif()
17
18 # find dependencies
19 find_package(ament_cmake REQUIRED)
20 find_package(rosidl_default_generators REQUIRED)
21 # uncomment the following section in order to fill in
22 # further dependencies manually.
23 # find_package(<dependency> REQUIRED)
24
25 rosidl_generate_interfaces(${PROJECT_NAME})
26   "action/Progress.action"
27   "msg/Person.msg"
28   "srv/Add.srv"
29 ]
30
31 if(BUILD_TESTING)
32   find_package(ament_lint_auto REQUIRED)
33   # the following line skips the linter which checks for copyrights
34   # uncomment the line when a copyright and license is not present in all source
35   #set(ament_cmake_copyright_FOUND TRUE)
36   # the following line skips cpplint (only works in a git repo)
37   # uncomment the line when this package is not in a git repo
38   #set(ament_cmake_cpplint_FOUND TRUE)

```

```

package.xml x
yahboomcar_ws > src > pkg_interfaces > package.xml
1 <?xml version="1.0"?>
2 <xmldoc href="http://download.ros.org/schema/package_format3.xsd" schematyp
3 <package format="3">
4   <name>pkg_interfaces</name>
5   <version>0.0.0</version>
6   <description>TODO: Package description</description>
7   <maintainer email="1461190907@qq.com">root</maintainer>
8   <license>TODO: License declaration</license>
9
10  <buildtool_depend>ament_cmake</buildtool_depend>
11
12  <test_depend>ament_lint_auto</test_depend>
13  <test_depend>ament_lint_common</test_depend>
14
15  <buildtool_depend>rosidl_default_generators</buildtool_depend>
16  <exec_depend>rosidl_default_runtime</exec_depend>
17  <depend>action_msgs</depend>
18  <member_of_group>rosidl_interface_packages</member_of_group>
19
20  <export>
21    <build_type>ament_cmake</build_type>
22  </export>
23 </package>

```

3、终端中进入当前工作空间，编译功能包：

```

cd ~/yahboomcar_ros2_ws/yahboomcar_ws
colcon build --packages-select pkg_interfaces
source install/setup.bash

```

4、测试

```

ros2 interface show pkg_interfaces/srv/Add

```

正常情况下，终端将会输出与 Person.msg 文件一致的内容。

```

root@unbutu:~/yahboomcar_ros2_ws/yahboomcar_ws# ros2 interface show pkg_interfaces/srv/Add
int32 num1
int32 num2
---
int32 sum
root@unbutu:~/yahboomcar_ros2_ws/yahboomcar_ws# 

```