



## ROS2命名规范

部 门 Division : ER  
版本号 Version : 1.0  
日 期 Date : 2019/11



# 目录

Catalog

01 > ROS2 rosidl\_adapter 命名规范

02 > 兼容ROS1命名规范修改指南

01



## ROS2 rosidl\_adapter 命名规范

## ➤ package命名规则：

1. 命名不能包含大写字母且必须以小写字母开始
2. 命名可以包含数字
3. 命名可以使用“\_”进行分割

例： `exAmple_package` ×

`1_example_package` ×

`example_package1` ✓

`example_package` ✓

## ➤ 文件命名规则：

1. 命名必须以大写字母开始
2. 命名可以包含数字
3. 命名不能包含“\_”

例： **vehicleControlCommand.msg** ×

**Vehicle\_Control\_Command.msg** ×

VehicleControlCommand.msg ✓

VehicleControlCommand2.msg ✓

➤ 每个变量包含变量类型和变量名两部分，且使用空格符进行分割。

➤ 变量命名规范：

1. 命名不能包含大写字母

例： float64 Variable float64 variAble ×

float64 variable √

2. 命名可包含数字但必须以小写字母开始，不能以“\_”结尾但可以使用“\_”进行分割

例： float64 3float\_msg float64 float\_msg\_ ×

float64 float\_msg float64 variable1 √

3. 对变量进行初始赋值

例： float64 float\_msg 6.66 √

string string\_msg “example “ √

int32[ ] samples [-1, 0, 1] √

### 4. 数组定义:

例: `int32[ ] array` // 无界整数数组

`int32[5] array` // 五维整数数组

`int32[<=5] array` // 最大维数5的整数数组

`string string_example` // 字符串长度不限

`string<=10 string_example` // 最多十个字符的字符串

`string[<=5] string_example` // 最多五个无界字符串

`string<=10[<=5] string_example` // 最多五个字符串，每个最多十个字符

## ➤ 常量命名规范

1. 命名不能出现小写字母
2. 命名可以包含数字但必须以大写字母开始
3. 命名可以使用 “\_” 进行分割

例: `int32 CONsT_INT = 11` ×

`int32 2CONST_INT = 11` ×

`int32 CONST_INT = 11` ✓

`int32 CONST_INT222 = 22` ✓

`string CONST_STRING = "example"` ✓





02



## 兼容ROS1命名规范修改指南

➤ 为了快速移植ROS1自定义msg/srv，可以通过修改源代码保证兼容性。修改步骤如下：

1. apt方式安装ROS2：

- `sudo gedit /opt/ros/xxx/lib/python3.6/site-packages/rosidl_adapter/parser.py`

2. 源码方式安装ROS2：

- `sudo gedit ros2_workspace/src/ros2/rosidl/rosidl_adapter/rosidl_adapter/parser.py`

## 3. 对parser.py中如下内容进行相应修改：

```
VALID_PACKAGE_NAME_PATTERN = re.compile('^[a-z]([a-z0-9_]?[a-z0-9]+)*$') // package 命名规范  
VALID_FIELD_NAME_PATTERN = re.compile('^[a-z]([a-z0-9_]?[a-z0-9]+)*$') // 变量命名规范  
VALID_MESSAGE_NAME_PATTERN = re.compile('^[A-Z][A-Za-z0-9]*$') // msg/srv文件名命名规范  
VALID_CONSTANT_NAME_PATTERN = re.compile('^[A-Z]([A-Z0-9_]?[A-Z0-9]+)*$') // 常量命名规范
```



```
VALID_PACKAGE_NAME_PATTERN = re.compile('^[A-Za-z]([a-z0-9_]?[a-z0-9]+)*$')  
VALID_FIELD_NAME_PATTERN = re.compile('^[A-Za-z]([A-Za-z0-9_]*$')  
VALID_MESSAGE_NAME_PATTERN = re.compile('^[A-Za-z][A-Za-z0-9_]*$')  
VALID_CONSTANT_NAME_PATTERN = re.compile('^[A-Z]([A-Z0-9_]?[A-Z0-9]+)*$')
```

上汽大众  
SAIC VOLKSWAGEN

