Robotics Competition 2021-22

eYRC 2021-22: Agri Bot (AB)

Create a srv file

Steps

1. Create a custom ROS Package or use any existing package like `pkg_ros_basics`.

2. Go into the package directory and create a folder called `srv`.

```
$ cd <package_name>
$ mkdir srv
```

3. 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` package:

```
$ roscp rospy_tutorials AddTwoInts.srv srv/AddTwoInts.srv
```

The srv file is,

```
int64 A
int64 B
---
int64 Sum
```

Here `A` and `B` holds the request data which is sent by the Client to the Server and `Sum` is the response which is sent by the Server to the Client.

4. Open `package.xml`, and make sure these two lines are in it and uncommented:

```
  <build_depend>message_generation</build_depend>
  <exec_depend>message_runtime</exec_depend>
```

5. 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
)
```

6. Remove `#` to uncomment the following lines:

```
# add_service_files(
#   FILES
#   Service1.srv
#   Service2.srv
# )
```

And replace the placeholder <service_name>.srv files for your service files:

```
add_service_files(
  FILES
  AddTwoInts.srv
)
```

7. Unless you have already done this in the previous steps, change in `CMakeLists.txt`.:

```
# generate_messages(
#   DEPENDENCIES
# #  std_msgs  # Or other packages containing msgs
# )
```

Uncomment it and add any packages you depend on which contain .msg files that your messages use (in this case std_msgs), such that it looks like this:

```
generate_messages(
  DEPENDENCIES
  std_msgs
)
```

8. Go to catkin_ws directory and run:

```
$ catkin_make
```

rossrv Command

The rossrv command-line tool displays information about ROS services. It has the exact same usage as rosmsg (see what it offers when it runs without sub-command below):

```
$ rossrv -h
```

```
rossrv is a command-line tool for displaying information about ROS Service types.

Commands:
        rossrv show     Show service description
        rossrv info     Alias for rossrv show
        rossrv list     List all services
        rossrv md5      Display service md5sum
        rossrv package  List services in a package
        rossrv packages List packages that contain services

Type rossrv <command> -h for more detailed usage
```