Create a action message file

- `.action` message files are the files which are used by ROS Actions for communication.
- `.action` files reside in `action` folder in a ROS package.

Steps

1. Let's first create a ROS Package called `pkg_ros_actions` that will depend on `rospy`, `roscpp`, `std_msgs` and **`actionlib_msgs`**.

```
cd ~/catkin_ws/src

catkin_create_pkg pkg_ros_actions roscpp rospy std_msgs actionlib_msgs
```

2. Now inside the package create a `action` folder for your `.action` file and inside which create a `myActionMsg.action`.

```
cd pkg_ros_actions

mkdir action

cd action

touch myActionMsg.action
```

3. Now open `myActionMsg.action` in your favourite text editor and fill it.

```
gedit myActionMsg.action
```

For this example we will have the following action definition,

```
# goal
int8 distance
int8 angle
---
# result
int8 final_x
int8 final_y
int8 final_theta
---
# feedback
int8 cur_x
int8 cur_y
int8 cur_theta
```

NOTE: If this action definition is used by Action Server and Client then,

- The Action Client will send a Goal which will have `distance` and `angle` of `int8` type to Action Server.

- The Action Server will send Feedback which will have `cur_x`, `cur_y` and `cur_theta` while it is processing the Goal.
- The Action Server will send Result which will have `final_x`, `final_y` and `final_theta` after processing the Goal.

4. Now in-order to make the catkin build system to generate action messages which can be used by ROS Nodes we would have to update `package.xml` and `CMakelist.txt`.

5. Open `package.xml` and add the following,

```
<!-- ** add message_generation as exec depend ** -->
  <exec_depend>message_generation</exec_depend>
```

Now your final `package.xml` should look something like this,

```
<?xml version="1.0"?>
<package format="2">

  <name>pkg_ros_actions</name>
  <version>0.0.0</version>
  <description>The pkg_ros_actions package</description>
  <maintainer email="eyantra@todo.todo">eyantra</maintainer>
  <license>TODO</license>

  <buildtool_depend>catkin</buildtool_depend>

  <build_depend>actionlib_msgs</build_depend>
  <build_depend>roscpp</build_depend>
  <build_depend>rospy</build_depend>
  <build_depend>std_msgs</build_depend>

  <build_export_depend>actionlib_msgs</build_export_depend>
  <build_export_depend>roscpp</build_export_depend>
  <build_export_depend>rospy</build_export_depend>
  <build_export_depend>std_msgs</build_export_depend>

  <exec_depend>actionlib_msgs</exec_depend>
  <exec_depend>roscpp</exec_depend>
  <exec_depend>rospy</exec_depend>
  <exec_depend>std_msgs</exec_depend>
  <exec_depend>message_generation</exec_depend>

  <export>
  </export>

</package>
```

6. Now update the `CMakelist.txt` of the package.

Add the following things,

```
## Generate actions in the 'action' folder
add_action_files(
  FILES
  myActionMsg.action
)

## Generate added messages and services with any dependencies listed here
```

```
generate_messages(
  DEPENDENCIES
  actionlib_msgscd
  std_msgs
)

catkin_package(
#  INCLUDE_DIRS include
#  LIBRARIES pkg_task1
  CATKIN_DEPENDS actionlib_msgs roscpp rospy std_msgs
#  DEPENDS system_lib
)
```

Now your `CMakelist.txt` should look like this,

```
cmake_minimum_required(VERSION 3.0.2)
project(pkg_ros_actions)


find_package(catkin REQUIRED COMPONENTS
  actionlib_msgs
  roscpp
  rospy
  std_msgs
)

## Generate actions in the 'action' folder
add_action_files(
  FILES
  myActionMsg.action
)

## Generate added messages and services with any dependencies listed here
generate_messages(
  DEPENDENCIES
  actionlib_msgs
  std_msgs
)

catkin_package(
#  INCLUDE_DIRS include
#  LIBRARIES pkg_ros_actions
 CATKIN_DEPENDS actionlib_msgs roscpp rospy std_msgs
#  DEPENDS system_lib
)

############
## Build ##
############

## Specify additional locations of header files
## Your package locations should be listed before other locations
include_directories(
# include
  ${catkin_INCLUDE_DIRS}
)
```

7. Now you are ready to build this package.

```
cd ~/catkin_ws

catkin build
```

```
source devel/setup.bash
```

8. Now if you go to `~/catkin_ws/devel/share/pkg_ros_actions/msg/` you will see all the generated messages with `.msg` extension which can be used for ROS Actions.

```
cd ~/catkin_ws/devel/share/pkg_ros_actions/msg/
```

OUTPUT

```
myActionMsgActionFeedback.msg  myActionMsgAction.msg        myActionMsgFeedback.msg
myActionMsgActionGoal.msg      myActionMsgActionResult.msg  myActionMsgGoal.msg
```

Recall that `.msg` files are used by ROS Topics. This tells us that ROS Actions internally uses ROS Topics for communication.

Now you can open `myActionMsgGoal.msg` to see its content.

```
cat myActionMsgGoal.msg
```

OUTPUT

```
# ====== DO NOT MODIFY! AUTOGENERATED FROM AN ACTION DEFINITION ======
# goal
int8 distance
int8 angle
```