Note: This tutorial assumes that you have completed the previous tutorials: ROS tutorials (/ROS/Tutorials).

🔦 Please ask about problems and questions regarding this tutorial on 🌐 answers.ros.org (http://answers.ros.org). Don't forget to include in your question the link to this page, the versions of your OS & ROS, and also add appropriate tags.

Defining Custom Messages

Description: This tutorial will show you how to define your own custom message data types using the ROS Message Description Language (/ROS/Message_Description_Language).

Tutorial Level:

Next Tutorial: Using a C++ class in Python (/ROS/Tutorials/Using%20a%20C%2B%2B %20class%20in%20Python)

catkin	rosbuild

1. Generating Messages

Generating a message is easy. Simply place a .msg file inside the msg directory in a package. Please follow previous tutorial about creating .msg files (/ROS/Tutorials/CreatingMsgAndSrv#Creating_a_msg) (don't forget to choose build system type at the top of the page there).

2. Including or Importing Messages

2.1 C++

Messages are put into a namespace that matches the name of the package. ie.

```
Afficher/masquer les numéros de lignes

1 #include <std_msgs/String.h>
2
3 std_msgs::String msg;
```

2.2 Python

Afficher/masquer les numéros de lignes

```
1 from std_msgs.msg import String
2
3 msg = String()
```

3. Dependencies

If you are using the new custom message defined in a different package, remember to add:

to package.xml (/catkin/package.xml):

```
<build_depend>name_of_package_containing_custom_msg</build_depend>
<exec_depend>name_of_package_containing_custom_msg</exec_depend>
```

and you will need to add this to your CMakeList.txt:

```
add_dependencies(your_program ${catkin_EXPORTED_TARGETS})
```

If you are building C++ nodes which use your new messages, you will also need to declare a dependency between your node and your message, as described in the 🌐catkin msg/srv build documentation (http://docs.ros.org/kinetic/api/catkin/html/howto/format2/cpp_msg_dependencies.html)

The ROSNodeTutorialPython (/ROSNodeTutorialPython) tutorial shows an example of the previously described talker and listener tutorials using a custom message, with implementations in C++ and Python.

Brought to you by: Open Source Robotics Foundation

(http://www.osrfoundation.org)