How to Define Custom Messages in ROS

Wyatt Newman, November, 2014

See http://wiki.ros.org/msg for details and background on ROS messages. This brief document shows how you can define more complex messages and message handlers. The specific example here is for sending and receiving a “path”, consisting of a variable-length list (a std::vector) of objects of Eigen-type Matrix, of fixed dimension 8x1 (i.e. an 8-DOF vector).

See the package “eigen_to_msg” in the class repository, under …/catkin/src/examples. In this package, two new message types are defined:

eigen_to_msg::vec_of_vec8dof

eigen_to_msg::vec8dof

These reside (as they must) in the package under a directory called “msg”.

The content of these message files, each of which ends with the suffix *.msg, is simple. For vec8dof.msg, the content is:

```
float64 a0
float64 a1
float64 a2
float64 a3
float64 a4
float64 a5
float64 a6
float64 a7
```

For messages of type eigen_to_msg::vec8dof, there will be components that may be accessed as *.a0, *.a1, etc. These are intended to hold 8 components of an 8-D vector.

The content of the vec_of_vec8dof.msg definition file is one line:

```
vec8dof[] vecs8dof
```

This means that a message of type eigen_to_msg::vec_of_vec8dof is a variable-length vector containing messages of type eigen_to_msg::vec8dof.

A header file, “eigen_to_msg.h”, is contained in the “include” directory of package “eigen_to_msg”. This header file contains four functions that perform the tasks of encoding and decoding messages to and from Eigen-type vectors (of fixed size 8x1) and variable-length lists (vectors) of these data types.

The example usage nodes “example_eigen_to_ros_subscriber.cpp” and “example_eigen_to_ros_publisher.cpp” show how to use these functions.

The message-definition files lead to creation of corresponding message header (*.h) files of the same name. This is performed by the build system. However, in the package.xml file, you must add (or uncomment) the lines:

```
 <build_depend>message_generation</build_depend>
```

and

```
<run_depend>message_runtime</run_depend>
```