catkin | rosbuild

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.

Creating a ROS package by hand.

Description: This tutorial explains how to manually create a ROS package.

Tutorial Level: INTERMEDIATE

Next Tutorial: Managing System Dependencies (/ROS/Tutorials/rosdep)

There is a tool for creating ROS Packages (/Packages) (catkin_create_pkg (/catkin/commands/catkin_create_pkg)), but, as you will see, there is nothing actually difficult here. `catkin_create_pkg` prevents mistakes and saves effort, but packages are just a directory and a simple XML file.

Now we'll create a new foobar package. This tutorial assumes that we're working your catkin workspace and sourcing of the setup file is already done.

```
catkin_ws_top $ mkdir -p src/foobar
catkin_ws_top $ cd src/foobar
```

The very first thing we'll do is add our manifest (/catkin/package.xml) file. The `package.xml` file allows tools like rospack (/rospack) to determine information about what your package depends upon.

Inside of `foobar/package.xml` put the following:

```
<package format="2">
  <name>foobar</name>
  <version>1.2.4</version>
  <description>
  This package provides foo capability.
  </description>
  <maintainer email="foobar@foo.bar.willowgarage.com">PR-foobar</maintainer>
  <license>BSD</license>

  <buildtool_depend>catkin</buildtool_depend>

  <build_depend>roscpp</build_depend>
  <build_depend>std_msgs</build_depend>

  <exec_depend>roscpp</exec_depend>
  <exec_depend>std_msgs</exec_depend>
</package>
```

See also this page from catkin tutorial (/catkin/Tutorials/CreatingPackage#ROS.2BAC8-Tutorials.2BAC8-catkin.2BAC8-CreatingPackage.Customizing_the_package.xml) for further information on catkin/package.xml (/catkin/package.xml).

Now that your package has a manifest, ROS can find it. Try executing the command:

```
rospack find foobar
```

If ROS is set up correctly you should see something like: `/home/user/ros/catkin_ws_top/src/foobar`. This is how ROS finds packages behind the scenes.

Note that this package now also has dependencies on roscpp (/roscpp) and std_msgs (/std_msgs).

Such dependencies are used by catkin to configure packages in the right order.

Now we need the CMakeLists.txt (/CMakeLists) file so that catkin_make (/catkin_make), which uses CMake for its more powerful flexibility when building across multiple platforms, builds the package.

In `foobar/CMakeLists.txt` put:

```
cmake_minimum_required(VERSION 2.8.3)
project(foobar)
find_package(catkin REQUIRED roscpp std_msgs)
catkin_package()
```

That's all you need to start building a package in ROS using `catkin`. Of course, if you want it to actually start building something, you're going to need to learn a couple more CMake macros. See our CMakeLists.txt (/catkin/CMakeLists.txt) guide for more information. Also always go back to beginner level tutorial (CreatingPackage (/ROS/Tutorials/CreatingPackage) and so on) to customize your `package.xml` and `CMakeLists.txt`.

Wiki: ROS/Tutorials/Creating a Package by Hand (dernière édition le 2018-05-21 18:40:34 par ChrisLalancette (/ChrisLalancette))

Brought to you by: Open Source Robotics Foundation

(http://www.osrfoundation.org)