ROS Compilation Steps

1. Create a directory using mkdir. This directory will be the workspace for ROS packages.
2. Create a sub-directory called src where the code will reside.
3. Inside the src directory packages can be created by the command:

catkin\_create\_pkg package-name

The command will create a directory structure for the package and will also create two files package.xml and CMakeLists.txt

1. CMakeLists.txt contains the build instructions and what all executables are needed for compilation of that particular package.
2. Package.xml contains which is not used anywhere else and can be used for documentation purpose.
3. Code for Helloworld
   1. ros/ros.h, argc, \*\*argv, ros::init(argc, argv, “hello\_ros”); ros::NodeHandle nh; ROS\_INFO\_STREAM(“Hello, ROS”);
4. Dependencies in CMakeLists.txt:
   1. Default will be find\_package(catkin REQUIRED)
   2. Should be changed to find\_package(catkin REQUIRED COMPONENTS package\_names)  Can have multiple packages.
   3. In our case it should be find\_package(catkin REQUIRED COMPONENTS roscpp)
   4. Replace

**include\_directories(**

**# include**

**# ${catkin\_INCLUDE\_DIRS}**

**)**

with

**include\_directories(**

**# include**

**${catkin\_INCLUDE\_DIRS}**

**)**

* 1. In CMakeLists.txt add the following lines:
     1. add\_executable(executable\_name souce\_files)
     2. target\_link\_libraries(executable\_name ${catkin\_LIBRARIES})
  2. In package.xml the following should be done:
     1. <build\_depend>package\_name</build\_depend>
     2. <exec\_depend>package\_name</exec\_depend>

1. Run the catkin\_make from the workspace directory.
2. All the projects will be built.
3. Then from the workspace directory run source devel/setup.bash
4. Then run roscore
5. Then run rosrun package\_name executable\_name