Back (/tutorials?cat=connect_ros)

ROS overview

Tutorial: ROS integration overview

To achieve ROS integration with stand-alone Gazebo, a set of ROS packages named gazebo_ros_pkgs (http://ros.org/wiki/gazebo_ros_pkgs) provides wrappers around the stand-alone Gazebo. They provide the necessary interfaces to simulate a robot in Gazebo using ROS messages, services and dynamic reconfigure Some features of `gazebo_ros_pkgs` :

- Supports a stand alone system dependency of Gazebo, that has no ROS bindings on its own
- Builds with catkin (http://www.ros.org/wiki/catkin)
- Treats URDF and SDF (http://gazebosim.org/sdf.html) as equally as possible
- Reduces code duplication with Gazebo
- Improves out of the box support for controllers using `ros_control`
- Integrates real time controller efficiency improvements from the DARPA Robotics Challenge
- Cleans up old code from previous versions of ROS and Gazebo

An overview of the `gazebo_ros_pkgs` interface is in the following diagram:

Upgrading from simulator_gazebo (ROS groovy and earlier)

The following guidelines will help you upgrade your Gazebo-dependent packages from `simulator_gazebo` for use in your ROS packages:

Launch Files

Some changes are required in previously created roslaunch files for starting Gazebo. The best way to update these packages is to review the Using roslaunch files to spawn models in Gazebo (http://gazebosim.org/tutorials?tut=ros_roslaunch&cat=connect_ros) tutorial. In a nutshell:

- Within roslaunch files, `pkg="gazebo"` needs to be now renamed to `pkg="gazebo_ros"`
- `gazebo_worlds` package has been removed. Most of the world files were rarely used and were not maintained with changes in SDF XML formats. Thus, all worlds have been centralized within the Gazebo project itself, including `empty.world` .
- The best way to use Gazebo launch files is to simply inherit/include the master `empty_world` launch file located in the `gazebo_ros` package.

CMakeLists.txt

The ROS-wrapped versiong of Gazebo was removed in favor of the system install of Gazebo. This may require reconfiguration of your CMake file. The following is an example CMakeLists.txt:

```
cmake_minimum_required(VERSION 2.8.3)
project(YOURROBOT_gazebo_plugins)

find_package(catkin REQUIRED COMPONENTS
  gazebo_ros
)

# Depend on system install of Gazebo
find_package(gazebo REQUIRED)

include_directories(include ${catkin_INCLUDE_DIRS} ${GAZEBO_INCLUDE_DIRS} ${SDFormat_INCLUDE_DIRS})

# Build whatever you need here
add_library(...) # TODO

catkin_package(
    DEPENDS
      gazebo_ros
    CATKIN_DEPENDS
    INCLUDE_DIRS
    LIBRARIES
)
```

package.xml

Add dependency on the new `gazebo_ros` package:

```
<build_depend>gazebo_ros</build_depend>
<run_depend>gazebo_ros</run_depend>
```

Running Gazebo

The names of the ROS nodes to launch Gazebo have changed slightly to coincide with the Gazebo executable names:

- `rosrun gazebo_ros gazebo` launch both the Gazebo server and GUI.
- `rosrun gazebo_ros gzclient` launch the Gazebo GUI.
- `rosrun gazebo_ros gzserver` launch the Gazebo server.

Available nodes to run:

```
rosrun gazebo_ros gazebo
rosrun gazebo_ros gzserver
rosrun gazebo_ros gzclient
rosrun gazebo_ros spawn_model
rosrun gazebo_ros perf
rosrun gazebo_ros debug
```

These nodes are better documented in the tutorial Using roslaunch files to spawn models in Gazebo (http://gazebosim.org/tutorials?tut=ros_roslaunch&cat=connect_ros).

Continue to Installing gazebo_ros Packages (http://gazebosim.org/tutorials?tut=ros_installing&cat=connect_ros).