Back (/tutorials?cat=connect_ros)

Installing gazebo_ros_pkgs

Version: 1.9+ Edit (https:/

Introduction

The set of ROS packages for interfacing with Gazebo are contained within a new meta package (catkin's version of stacks) named `gazebo_ros_pkgs` . See Overview of new ROS integration (http://gazebosim.org/tutorials/?tut=ros_overview) for background information before continuing here.

These instructions are for using the Gazebo versions that are fully integrated with ROS Kinetic (http://www.ros.org/wiki/kinetic), ROS Jade (http://www.ros.org/wiki/jade) and ROS Indigo (http://www.ros.org/wiki/indigo). It is recommended to first read Which combination of ROS/Gazebo version to use (http://gazebosim.org/tutorials/?tut=ros_wrapper_versions) before going on with this tutorial. Depending on your needs, you could need an alternative installation.

Prerequisites

You should understand the basic concepts of ROS and have gone through the ROS Tutorials (http://www.ros.org/wiki/ROS/Tutorials).

Install ROS

We recommend for these ROS integration tutorials you install (`ros-kinetic-desktop-full` , `ros-jade-desktop-full` or `ros-indigo-desktop-full`) so that you have all the necessary packages.

See the ROS installation page (http://www.ros.org/wiki/ROS/Installation) for more details. Be sure to source your ROS setup.bash script by following the instructions on the ROS installation page.

Install Gazebo

You can install Gazebo either from source or from pre-build Ubuntu debians.

See Install Gazebo (http://gazebosim.org/tutorials?cat=install). If installing from source, be sure to build the `gazebo_X.Y` (X.Y being your desired version) branch.

Test that stand-alone Gazebo works

Before attempting to install the `gazebo_ros_pkgs` , make sure the stand-alone Gazebo works by running in terminal:

```
gazebo
```

You should see the GUI open with an empty world. Also, test adding a model by clicking on the "Insert" tab on the left and selecting a model to add (then clicking on the simulation to select where to place the model).

Test that you have the right version of Gazebo

To see where you install Gazebo, and if it is in the correct location, run:

```
which gzserver
which gzclient
```

If you installed from source to the default location it should say:

```
Do not copy. The following is for reference purposes.
```

```
/usr/local/bin/gzserver
/usr/local/bin/gzclient
```

If you installed from debian it should say:

```
Do not copy. The following is for reference purposes.
```

```
/usr/bin/gzserver
/usr/bin/gzclient
```

Install gazebo_ros_pkgs

Choose the method you would prefer. The easier and faster is installing it from packages but installing from source means you can more easily debug and submit bug patches ;-)

A. Install Pre-Built Debians

The `gazebo_ros_pkgs` packages are available in:

- ROS Kinetic (http://ros.org/wiki/kinetic):

```
sudo apt-get install ros-kinetic-gazebo-ros-pkgs ros-kinetic-gazebo-ros-control
```

- ROS Jade (http://ros.org/wiki/jade):

```
sudo apt-get install ros-jade-gazebo-ros-pkgs
```

- ROS Indigo (http://ros.org/wiki/indigo):

```
sudo apt-get install ros-indigo-gazebo-ros-pkgs ros-indigo-gazebo-ros-control
```

If this installation method ends successfully for you, jump to the Testing Gazebo with ROS Integration section below.

B. Install from Source (on Ubuntu)

If you are running an earlier version of ROS (Groovy or earlier) you will need to install `gazebo_ros_pkgs` from source. Installing from source is also useful if you want to develop new plugins or submit patches.

Setup A Catkin Workspace

These instructions require the use of the catkin (http://www.ros.org/wiki/catkin) build system.

If you do not have a catkin workspace setup, try the following commands:

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/src
catkin_init_workspace
cd ~/catkin_ws
catkin_make
```

Then add to your `.bashrc` file a source to the setup scripts:

```
echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
```

For more details see the Create A Catkin Workspace (http://www.ros.org/wiki/catkin/Tutorials/create_a_workspace) tutorial.

Clone the Github Repositories

Make sure `git` is installed on your Ubuntu machine:

```
sudo apt-get install git
```

ROS Kinetic

Kinetic is using the gazebo 7.x series, start by installing it:

```
sudo apt-get install -y libgazebo7-dev
```

Download the source code from the `gazebo_ros_pkgs` github repository (https://github.com/ros-simulation/gazebo_ros_pkgs):

```
cd ~/catkin_ws/src
git clone https://github.com/ros-simulation/gazebo_ros_pkgs.git -b kinetic-devel
```

Check for any missing dependencies using rosdep:

```
rosdep update
rosdep check --from-paths . --ignore-src --rosdistro kinetic
```

You can automatically install the missing dependencies using rosdep via debian install:

```
rosdep install --from-paths . --ignore-src --rosdistro kinetic -y
```

Now jump to the build the gazebo_ros_pkgs section.

ROS Jade

Jade is using the gazebo 5.x series, start by installing it:

```
sudo apt-get install -y libgazebo5-dev
```

Download the source code from the `gazebo_ros_pkgs` github repository (https://github.com/ros-simulation/gazebo_ros_pkgs):

```
cd ~/catkin_ws/src
git clone https://github.com/ros-simulation/gazebo_ros_pkgs.git -b jade-devel
```

Check for any missing dependencies using rosdep:

```
rosdep update
rosdep check --from-paths . --ignore-src --rosdistro jade
```

You can automatically install the missing dependencies using rosdep via debian install:

```
rosdep install --from-paths . --ignore-src --rosdistro jade -y
```

Note: currently in ROS Jade there is no ros-jade-gazebo-ros-control package released. Check the issue in the gazebo_ros_control tracker (https://github.com/ros-controls/ros_control/issues/201) to see the progress. Meantime, we need to disable the gazebo-ros-control compilation:

```
touch gazebo_ros_pkgs/gazebo_ros_control/CATKIN_IGNORE
```

Now jump to the build the gazebo_ros_pkgs section.

ROS Indigo

Indigo is using the gazebo 2.x series, start by installing it:

```
sudo apt-get install -y gazebo2
```

Download the source code from the `gazebo_ros_pkgs` github repository (https://github.com/ros-simulation/gazebo_ros_pkgs):

```
cd ~/catkin_ws/src
git clone https://github.com/ros-simulation/gazebo_ros_pkgs.git -b indigo-devel
```

Check for any missing dependencies using rosdep:

```
rosdep update
rosdep check --from-paths . --ignore-src --rosdistro indigo
```

You can automatically install the missing dependencies using rosdep via debian install:

```
rosdep install --from-paths . --ignore-src --rosdistro indigo -y
```

Now jump to the build the gazebo_ros_pkgs section.

Build the gazebo_ros_pkgs

To build the Gazebo ROS integration packages, run the following commands:

```
cd ~/catkin_ws/
catkin_make
```

See answers.gazebosim.org (http://answers.gazebosim.org/questions/) for issues or questions with building these packages.

Testing Gazebo with ROS Integration

Be sure to always source the appropriate ROS setup file, which for Kinetic is done like so:

```
source /opt/ros/kinetic/setup.bash
```

You might want to add that line to your `~/.bashrc` .

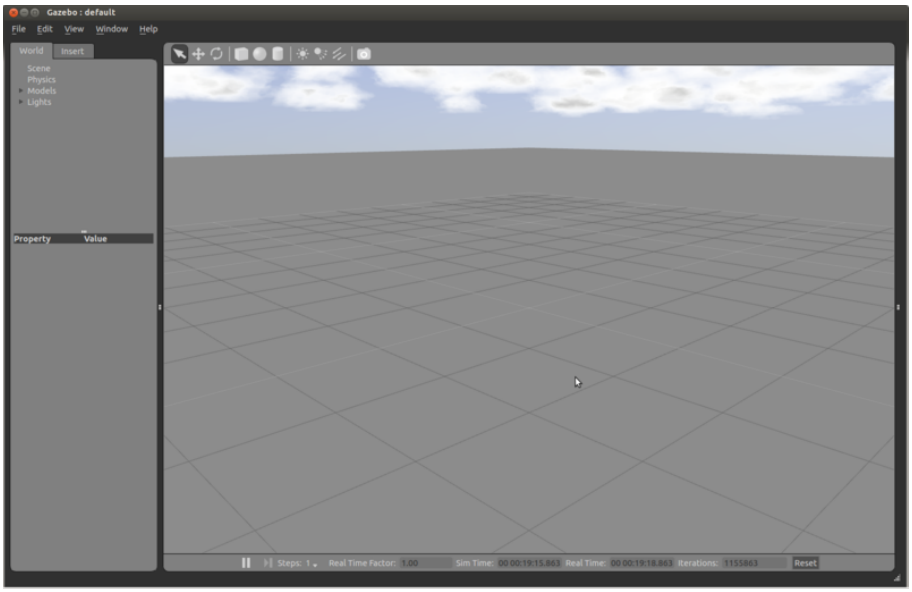
Assuming your ROS and Gazebo environment have been properly setup and built, you should now be able to run Gazebo through a simple `rosrun` command, after launching `roscore` if needed:

Source the catkin setup.bash if it's not already in your .bashrc

```
source ~/catkin_ws/devel/setup.bash
```

```
roscore &
rosrun gazebo_ros gazebo
```

The Gazebo GUI should appear with nothing inside the viewing window.



To verify that the proper ROS connections are setup, view the available ROS topics:

```
rostopic list
```

You should see within the lists topics such as:

```
Do not copy. The following is for reference purposes.
```

```
/gazebo/link_states
/gazebo/model_states
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/set_link_state
/gazebo/set_model_state
```

You can also verify the Gazebo services exist:

```
rosservice list
```

You should see within the list services such as:

```
Do not copy. The following is for reference purposes.
```

```
/gazebo/apply_body_wrench
/gazebo/apply_joint_effort
/gazebo/clear_body_wrenches
/gazebo/clear_joint_forces
/gazebo/delete_model
/gazebo/get_joint_properties
/gazebo/get_link_properties
/gazebo/get_link_state
/gazebo/get_loggers
/gazebo/get_model_properties
/gazebo/get_model_state
/gazebo/get_physics_properties
/gazebo/get_world_properties
/gazebo/pause_physics
/gazebo/reset_simulation
/gazebo/reset_world
/gazebo/set_joint_properties
/gazebo/set_link_properties
/gazebo/set_link_state
/gazebo/set_logger_level
/gazebo/set_model_configuration
/gazebo/set_model_state
/gazebo/set_parameters
/gazebo/set_physics_properties
/gazebo/spawn_gazebo_model
/gazebo/spawn_sdf_model
/gazebo/spawn_urdf_model
/gazebo/unpause_physics
/rosout/get_loggers
/rosout/set_logger_level
```

Other ROS Ways To Start Gazebo

There are several `rosrun` commands for starting Gazebo:

- Launch both the server and client together

```
rosrun gazebo_ros gazebo
```

- Launch the Gazebo server only

```
rosrun gazebo_ros gzserver
```

- Launch the Gazebo client only

```
rosrun gazebo_ros gzclient
```

- Launches the Gazebo server only, in debug mode using GDB

```
rosrun gazebo_ros debug
```

- Additionally, you can start Gazebo using `roslaunch`

```
roslaunch gazebo_ros empty_world.launch
```