
XACRO Basics

Estimated time to completion: **5 minutes**

7.4 Process XACRO Files inside Launch Files

XACRO files cannot be directly used in robot simulators as they are not compatible with the simulators. They must first be processed and converted into URDF files to use them in simulation. This conversion process can be triggered from inside a launch file when the robot model is to be spawned in a specific robot simulator such as Gazebo.

The **robot_state_publisher** node is often used to load these converted files into Gazebo. This node is responsible for publishing the robot model description into the ROS network. Then, `spawn_entity.py` can use this information to create a robot model instance in Gazebo to generate a URDF file for a given robot model.

Below is an exemplary launch file showing how to process an XACRO file, convert it into a URDF robot description, and pass it to the `robot_state_publisher` node.

► Execute in Webshell 1

```
In [ ]: cd ~/ros2_ws/src/my_box_bot_description/launch
```



```
In [ ]: touch box_bot_xacro.launch.py
```



Use an IDE to write the contents of the file, as shown in the example below:

 `box_bot_xacro.launch.py`

In []:



```
import os
from ament_index_python.packages import get_package_share_directory
from launch import LaunchDescription
from launch.actions import ExecuteProcess, IncludeLaunchDescription
from launch.launch_description_sources import PythonLaunchDescriptionSource
from launch_ros.actions import Node
from ament_index_python.packages import get_package_prefix

import xacro

def generate_launch_description():

    pkg_box_bot_gazebo = get_package_share_directory('my_box_bot_gazebo')
    description_package_name = "my_box_bot_description"
    install_dir = get_package_prefix(description_package_name)

    # This is to find the models inside the models folder in my_box_bot_gazebo package
    gazebo_models_path = os.path.join(pkg_box_bot_gazebo, 'models')
    if 'GAZEBO_MODEL_PATH' in os.environ:
        os.environ['GAZEBO_MODEL_PATH'] = os.environ['GAZEBO_MODEL_PATH'] + \
            ':' + install_dir + '/share' + ':' + gazebo_models_path
    else:
        os.environ['GAZEBO_MODEL_PATH'] = install_dir + \
            "/share" + ':' + gazebo_models_path

    if 'GAZEBO_PLUGIN_PATH' in os.environ:
        os.environ['GAZEBO_PLUGIN_PATH'] = os.environ['GAZEBO_PLUGIN_PATH'] + \
            ':' + install_dir + '/lib'
    else:
        os.environ['GAZEBO_PLUGIN_PATH'] = install_dir + '/lib'

    print("GAZEBO MODELS PATH==" + str(os.environ["GAZEBO_MODEL_PATH"]))
    print("GAZEBO PLUGINS PATH==" + str(os.environ["GAZEBO_PLUGIN_PATH"]))

    gazebo = IncludeLaunchDescription(
        PythonLaunchDescriptionSource([os.path.join(
```

```

        get_package_share_directory('gazebo_ros'), 'launch'), '/gazebo.launch.py']],
        launch_arguments={"verbose": "false", 'pause': 'true'}.items(),
    )

    robot_model_path = os.path.join(
        get_package_share_directory('my_box_bot_description'))

    xacro_file = os.path.join(robot_model_path, 'urdf', 'box_bot.xacro')

    # convert XACRO file into URDF
    doc = xacro.parse(open(xacro_file))
    xacro.process_doc(doc)
    params = {'robot_description': doc.toxml()}

    robot_state_publisher = Node(
        package='robot_state_publisher',
        executable='robot_state_publisher',
        output='screen',
        parameters=[params]
    )

    spawn_entity = Node(package='gazebo_ros', executable='spawn_entity.py',
                        arguments=['-entity', 'my_box_bot', '-x', '1.0', '-y', '1.0', '-z', '0.2',
                                   '-topic', 'robot_description'],
                        output='screen')

    return LaunchDescription([
        gazebo,
        robot_state_publisher,
        spawn_entity,
    ])

```

► Execute in Webshell 1

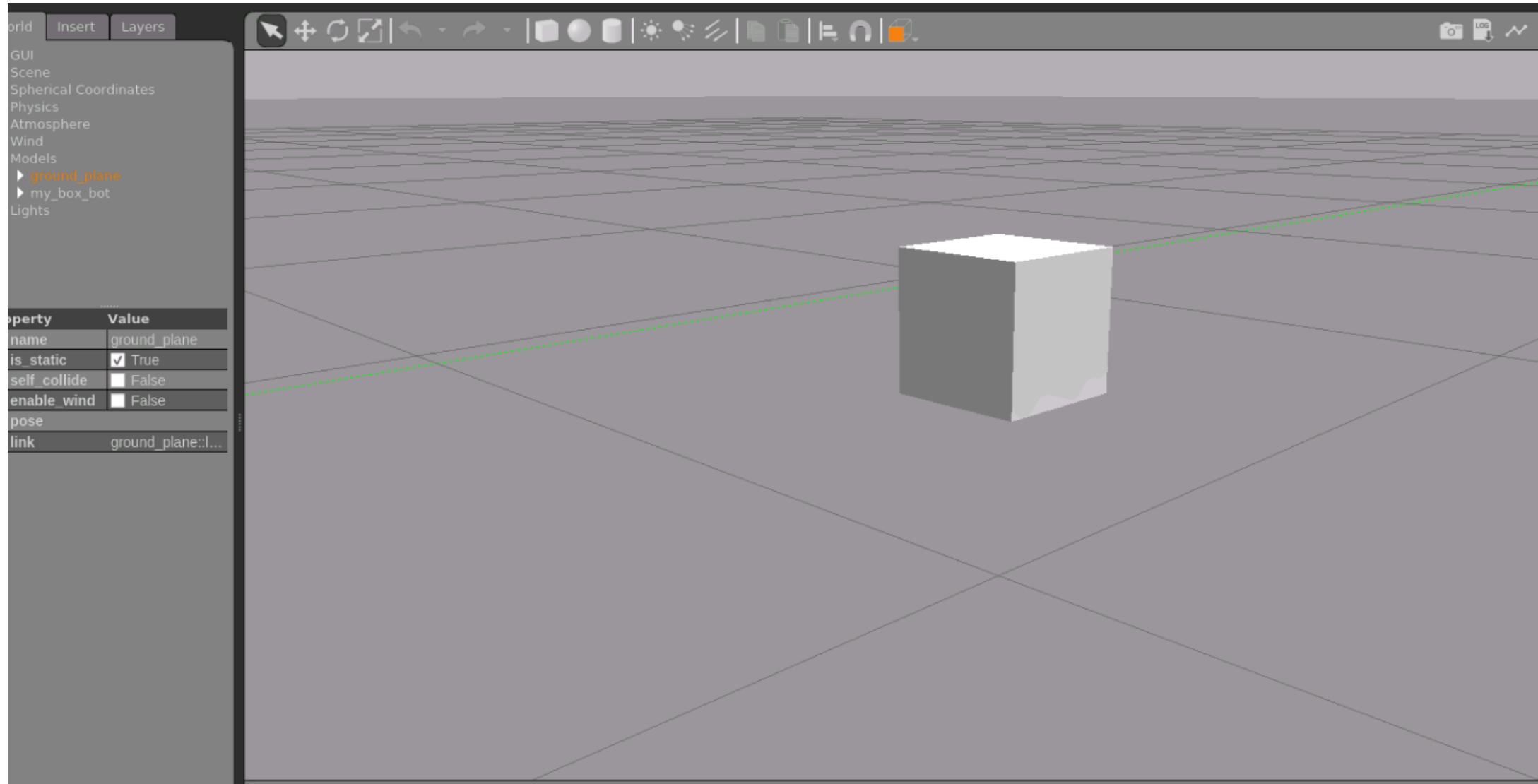
In []: `cd ~/ros2_ws && colcon build && source install/setup.bash`



In []: `ros2 launch my_box_bot_description box_bot_xacro.launch.py`



At this point in the unit, the `box_bot.xacro` robot model consists only of a cube with dimensions of 0.1 x 0.1 x 0.1 meters. This is what you should see in the simulation. You can terminate all processes in the terminal where you executed the launch file as they are no longer needed.





16/11/2023