

# AI & Robotics

ROS next steps

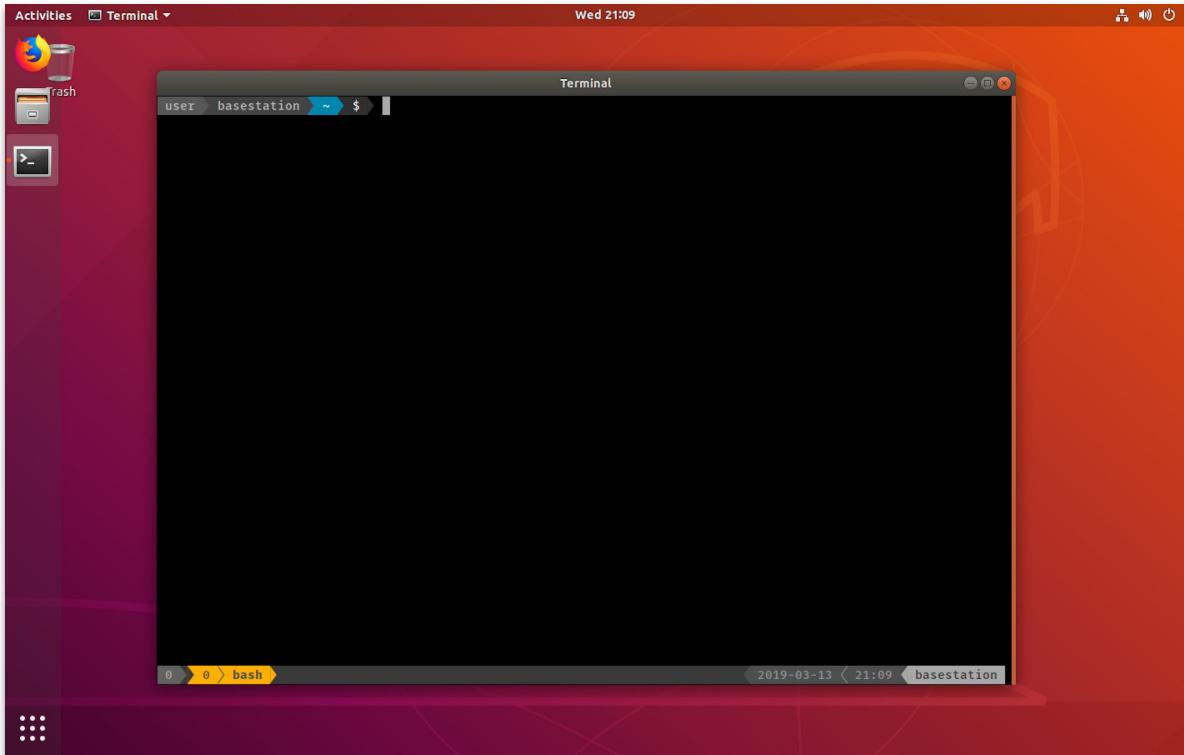


# Goals

**This part has no new goals.** It contains links to supplementary study materials for previously defined goals and extra information.



# tmux: terminal multiplexer (extra)



# tmux: a tutorial (extra)

HOME ABOUT ADVISORY BLOG

## Tmux Tutorial: An Easy Guide with Screenshots and Examples (2018 Update)

*There were many times I wished I could run a script on a VPS without staying connected to the instance via SSH. You could use Tmux to achieve this.*

2018,Jun 14 ⌂ Edric Teo

---

If you are a computer science student or someone who enjoys working on a VPS, chances are you are using Terminal 90% of the time.

Unless you are working on UI using terminal, mastering some terminal commands bring about some benefits.

Maybe it's just me. But ever since being comfortable with terminal commands, I realised that I had slowly shift away from UI.

Or maybe I get the feeling of being a l33t hacker getting simple operations done using the terminal.

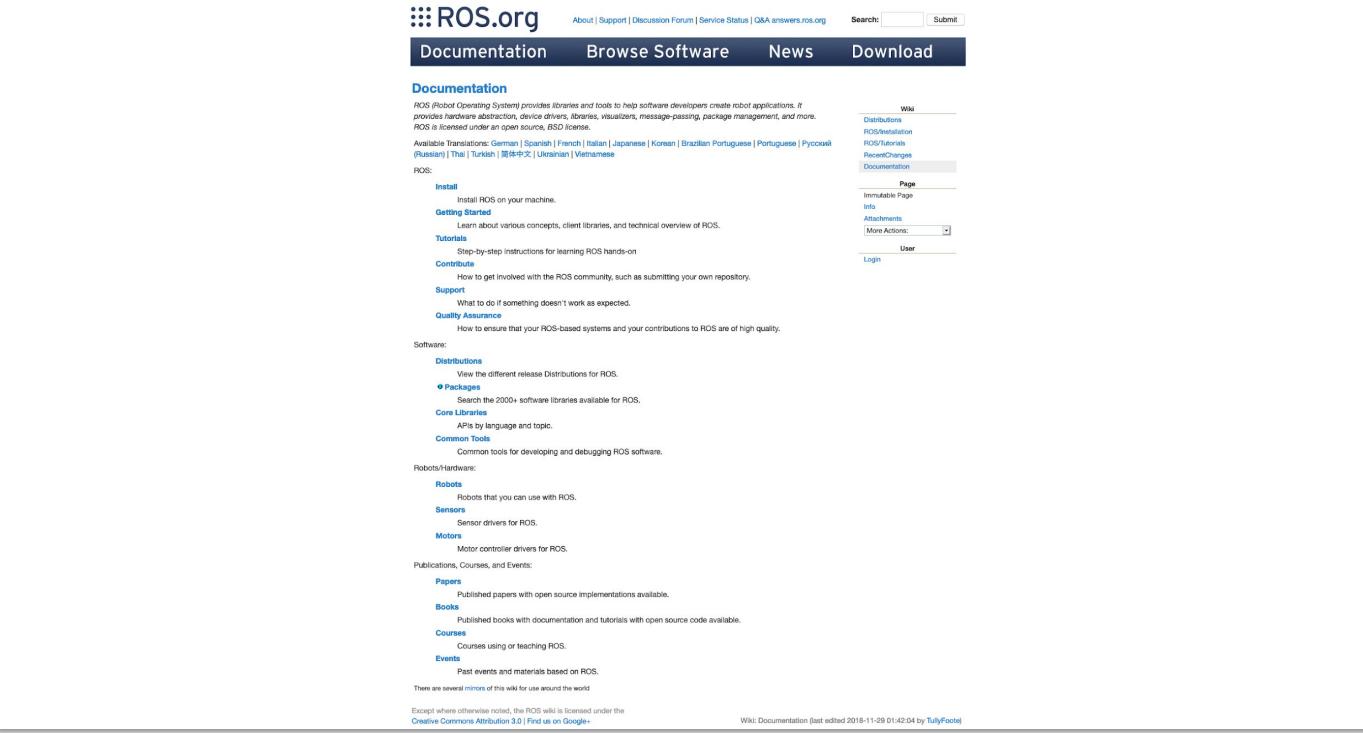
While working on VPS, there was always a problem of wanting to run a script on it without maintaining a SSH connection.

In addition, terminals that are in full screen usually utilises only the half

[SOURCE]

<https://edricteo.com/tmux-tutorial>

# ROS Documentation



The screenshot shows the ROS.org Documentation page. At the top, there is a navigation bar with links for About, Support, Discussion Forum, Service Status, and Q&A answers. A search bar is also present. Below the navigation bar, there are five main menu items: Documentation, Browse Software, News, and Download. The Documentation menu is currently selected.

The main content area is titled "Documentation". It contains several sections:

- ROS:** Includes links for Install, Getting Started, Tutorials, Contribute, Support, and Quality Assurance.
- Software:** Includes links for Distributions, Packages, Core Libraries, Common Tools, and Robots.
- Robotics/Hardware:** Includes links for Robots, Sensors, Motors, and Publications, Courses, and Events.
- Papers:** Published papers with open source implementations available.
- Books:** Published books with documentation and tutorials with open source code available.
- Courses:** Courses using or teaching ROS.
- Events:** Past events and materials based on ROS.

On the right side of the page, there is a sidebar with the following sections:

- Wiki:** Includes links for Distributions, ROS/testflight, ROS/Tutorials, RecentChanges, and Documentation.
- Page:** Includes links for Immutable Page, etc., and Attachments.
- User:** Includes a "More Actions" dropdown and a Login link.

At the bottom of the page, there is a note about mirrors and a Creative Commons Attribution 3.0 license notice. The footer also includes a link to the Wiki Documentation page.

[SOURCE]

<http://wiki.ros.org>

# Installation

[About](#) | [Support](#) | [Discussion Forum](#) | [Service Status](#) | [Q&A answers.ros.org](#)

Search:

[Documentation](#)   [Browse Software](#)   [News](#)   [Download](#)

[ROS/ Installation](#)

## ROS Installation Options

There is more than one ROS distribution supported at a time. Some are older releases with long term support, making them more stable, while others are newer with shorter support life times, but with binaries for more recent platforms and more recent versions of the ROS packages that make them up. See the [Distributions](#) page for more details. We recommend one of the versions below:

<p><b>ROS Kinetic Kame</b> Released May, 2016 LTS, supported until April, 2021</p> 	<p><b>ROS Melodic Morenia</b> Released May, 2018 Latest LTS, supported until May, 2023</p> 
--	--

Except where otherwise noted, the ROS wiki is licensed under the [Creative Commons Attribution 3.0](#) | [Find us on Google+](#)

Wiki: ROS/Installation (last edited 2018-06-07 21:24:33 by Marguedas)

Brought to you by:  Open Source Robotics Foundation

[SOURCE]

<http://wiki.ros.org/ROS/Installation>

# Installation

## 1. Installation

### 1.1 Configure your Ubuntu repositories

Configure your Ubuntu repositories to allow "restricted," "universe," and "multiverse." You can [follow the Ubuntu guide](#) for instructions on doing this.

### 1.2 Setup your sources.list

Setup your computer to accept software from packages.ros.org.

```
sudo sh -c "echo 'deb http://packages.ros.org/ros/ubuntu $ROS_DISTRO main' > /etc/apt/sources.list.d/roslatest.list"
```

[Mirrors](#) Source Debs are also available

### 1.3 Set up your keys

```
sudo apt-key adv --keyserver https://keyserver.ubuntu.com:443 --recv-key 421C365B9FF1F717815A5A95373A8E8019A116
```

If you experience issues connecting to the keyserver, you can try substituting <http://pgp.mit.edu:80> or <http://keyserver.ubuntu.com:80> in the previous command.

### 1.4 Installation

First, make sure your Debian package index is up-to-date:

```
sudo apt update
```

There are many different libraries and tools in ROS. We provided four default configurations to get you started. You can also install ROS packages individually.

In case of problems with the next step, you can use following repositories instead of the ones mentioned above ([ros-shadowed](#)):

**Debian-Full Install: (Recommended)**: ROS, [rqt](#), [rviz](#), robot-generic libraries, 2D/3D simulators and 2D/3D perception

```
sudo apt install ros-melodic-desktop-full
```

or [click here](#)

**Desktop Install**: ROS, [rqt](#), [rviz](#), and robot-generic libraries

```
sudo apt install ros-melodic-desktop
```

or [click here](#)

**ROS-Bare**: (Bare Bones) ROS package, build, and communication libraries. No GUI tools.

```
sudo apt install ros-melodic-ros-base
```

or [click here](#)

**Individual Package**: You can also install a specific ROS package (replace underscores with dashes of the package name):

```
sudo apt install ros-melodic-PACKAGE
```

e.g.

```
sudo apt install ros-melodic-slam-gmapping
```

To find available packages, use:

```
apt search ros-melodic
```

### 1.5 Initialize rosdep

Before you can use ROS, you will need to initialize rosdep. rosdep enables you to easily install system dependencies for source you want to compile and is required to run some core components in ROS.

```
sudo rosdep init
```

[SOURCE]

<http://wiki.ros.org/melodic/Installation/Ubuntu>

# Tutorials

The screenshot shows the ROS.org website with the URL <http://ros.org>. The main navigation bar includes links for About, Support, Discussion Forum, Service Status, and Q&A answers. A search bar is at the top right. Below the header, there are tabs for Documentation, Browse Software, News, and Download. The current page is ROS/Tutorials, which displays the ROS Tutorials content. On the left, there's a sidebar with a table of contents for ROS Tutorials, including sections like Core ROS Tutorials, Beginner Level, and Installation. The main content area features a heading for "ROS Tutorials" and a paragraph about non-beginner users. It then lists 11 numbered tutorials under "1. Core ROS Tutorials". Each tutorial has a brief description and a link to its full page. The right side of the page includes a sidebar with links for Wiki, Distributions, ROS/Installation, ROS/Tutorials, RecentChanges, Page, Immutable Page, Info, Attachments, and More Actions. There's also a "User" section with a "Login" link.

## ROS Tutorials

**Non-Beginners:** If you're already familiar enough with `ros fuerte` or earlier versions and only want to explore the new build system introduced in `groovy` and used in `hydro` and later, called `catkin`, you can go through more in-depth [catkin tutorial](#) here. However, going over all basic [Beginner Level](#) tutorials is still recommended for all users to get exposed to new features.

If you are new to Linux, you may find it helpful to first do a quick tutorial on common command line tools for Linux. A good one is [this here](#).

### 1. Core ROS Tutorials

#### 1.1 Beginner Level

1. [Installing and Configuring Your ROS Environment](#)  
This tutorial walks you through installing ROS and setting up the ROS environment on your computer.
2. [Navigating the ROS Filesystem](#)  
This tutorial introduces ROS filesystem concepts, and covers using the `rosmake`, `roslaunch`, and `rospack` commandline tools.
3. [Creating a ROS Package](#)  
This tutorial covers using `roscreate-pkg` or `catkin` to create a new package, and `rospack` to list package dependencies.
4. [Building a ROS Package](#)  
This tutorial covers the toolchain to build a package.
5. [Understanding ROS Nodes](#)  
This tutorial introduces ROS graph concepts and discusses the use of `roscore`, `rosnode`, and `rosrun` commandline tools.
6. [Understanding ROS Topics](#)  
This tutorial introduces ROS topics as well as using the `rostopic` and `rqt_plot` commandline tools.
7. [Understanding ROS Services and Parameters](#)  
This tutorial introduces ROS services, and parameters as well as using the `rosservice` and `rosparam` commandline tools.
8. [Using `rqt\_console` and `roslaunch`](#)  
This tutorial covers using `rqt_console` and `rqt_logger_level` for debugging and `roslaunch` for starting many nodes at once. If you use `ros fuerte` or earlier distros where `rqt` isn't fully available, please see this page with [this](#) page that uses old `rx` based tools.
9. [Using `roscd` to edit files in ROS](#)  
This tutorial shows how to use `roscd` to make editing easier.
10. [Creating a ROS msg and srv](#)  
This tutorial covers how to create and build msg and srv files as well as the `rosmsg`, `rossrv` and `roscpp` commandline tools.
11. [Writing a Simple Publisher and Subscriber \(C++\)](#)

[SOURCE]

<http://wiki.ros.org/ROS/Installation>

# Creating a workspace for catkin

The screenshot shows a screenshot of the ROS.org website. At the top, there is a navigation bar with links for 'About', 'Support', 'Discussion Forum', 'Service Status', and 'Q&A answers.ros.org'. There is also a search bar and a 'Submit' button. Below the navigation bar, there are four main menu items: 'Documentation', 'Browse Software', 'News', and 'Download'. The 'Documentation' item is currently selected. In the center, there is a title 'catkin/ Tutorials/ create\_a\_workspace' followed by a note: 'Please ask about problems and questions regarding this tutorial on [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.' Below this note, the main title of the tutorial is 'Creating a workspace for catkin'. A 'Description' section states: 'This tutorial covers how to setup a catkin workspace in which one or more catkin packages can be built.' A 'Keywords' section lists 'catkin workspace'. A 'Tutorial Level' section indicates it is for 'BEGINNER'. A 'Next Tutorial' section links to 'Creating catkin packages'. Below the main content, there is a row of buttons for different ROS distributions: 'indigo', 'kinetic', 'lunar', and 'melodic' (which is highlighted). There is also a checkbox for 'Show EOL distros'. At the bottom of the page, there is a 'Contents' section with a link to '1. Prerequisites'. The page also includes a sidebar with links for 'Wiki', 'Distributions', 'ROS/Installation', 'ROS/Tutorials', 'RecentChanges', and 'create\_a\_workspace' (which is highlighted). Other sidebar links include 'Page', 'Immutable Page', 'Info', 'Attachments', 'More Actions', and 'User' (with a 'Login' link).

## 1. Prerequisites

This tutorial assumes that you have [installed catkin](#) and sourced your environment. If you installed catkin via apt-get for ROS melodic, your command would look like this:

```
$ source /opt/ros/melodic/setup.bash
```

Let's create and build a [catkin workspace](#):

[SOURCE]

[http://wiki.ros.org/catkin/Tutorials/create\\_a\\_workspace](http://wiki.ros.org/catkin/Tutorials/create_a_workspace)

# Gazebo

The screenshot shows the official Gazebo website. At the top, there's a navigation bar with links to Tutorials, Download, Blog, Media, and Projects. A search bar is also present. The main header features the Gazebo logo (an orange cube icon) and the tagline "Robot simulation made easy." Below the header, there are download links for "Download (v0.0)" and "View on Bitbucket". A "GET STARTED" button is also visible. To the right, there are links for "SDFormat : Robot and simulation model format" and "Ignition : Libraries for robot applications". Social media sharing icons for Google+, Facebook, and Twitter are located at the bottom right of the header area.

**Why Gazebo?**

Robot simulation is an essential tool in every robotics toolbox. A well-designed simulator makes it possible to rapidly test algorithms, design robots, perform regression testing, and train AI system using realistic scenarios. Gazebo offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. It includes a robust robot physics engine, high-quality graphics, and convenient programmatic and graphical interfaces. Best of all, Gazebo is free with a vibrant community.

**The Latest**

**Gazebo Newsletter: November 2018**  
2018-11-05

**Gazebo Newsletter 14**  
November 2018

Libraries within the Ignition Robotics framework have been under development for a few years now. For example, Ignition Transport was started back in 2014. Ignition Robotics is now ready.

**Useful Links**

- Answers**  
Find answers and ask questions.
- Community**  
Join for discussions and announcements.
- Simulation Models**  
Robots, objects, and other simulation models.
- Source code**  
Get Gazebo's source code.

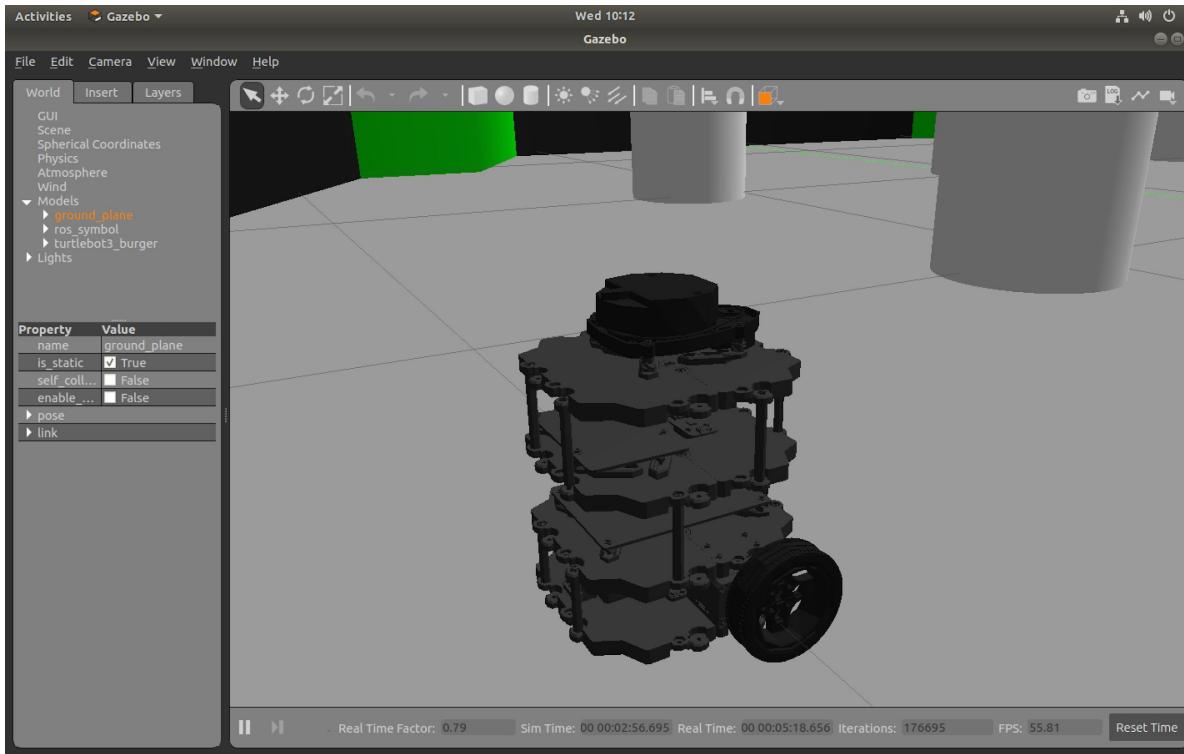
**Features**

Dynamics Simulation Access multiple high-performance physics engines including ODE, Bullet, Simbody, and DART.	Advanced 3D Graphics Utilizing OGRE, Gazebo provides realistic rendering of environments including high-quality lighting, shadows, and textures.	Sensors and Noise Generate sensor data, optionally with noise, from laser range finders, 2D/3D cameras, Kinect style sensors, contact sensors, force-torque, and more.	Plugins Develop custom plugins for robot, sensor, and environmental control. Plugins provide direct access to Gazebo's API.
Robot Models	TCP/IP Transport	Cloud Simulation	Command Line Tools

[SOURCE]

<http://gazebosim.org>

# Turtlebot 3 burger in Gazebo



# Turtlebot 3 burger in Gazebo

```
$ sudo apt-get install ros-melodic-turtlebot3*
$ roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch
... logging to /home/user/.ros/log/5be70128-4595-11e9-9fba-000c29e9707b/roslaunch-basestation-19328.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
```

**RLEexception: Invalid <arg> tag: environment variable 'TURTLEBOT3\_MODEL' is not set.**

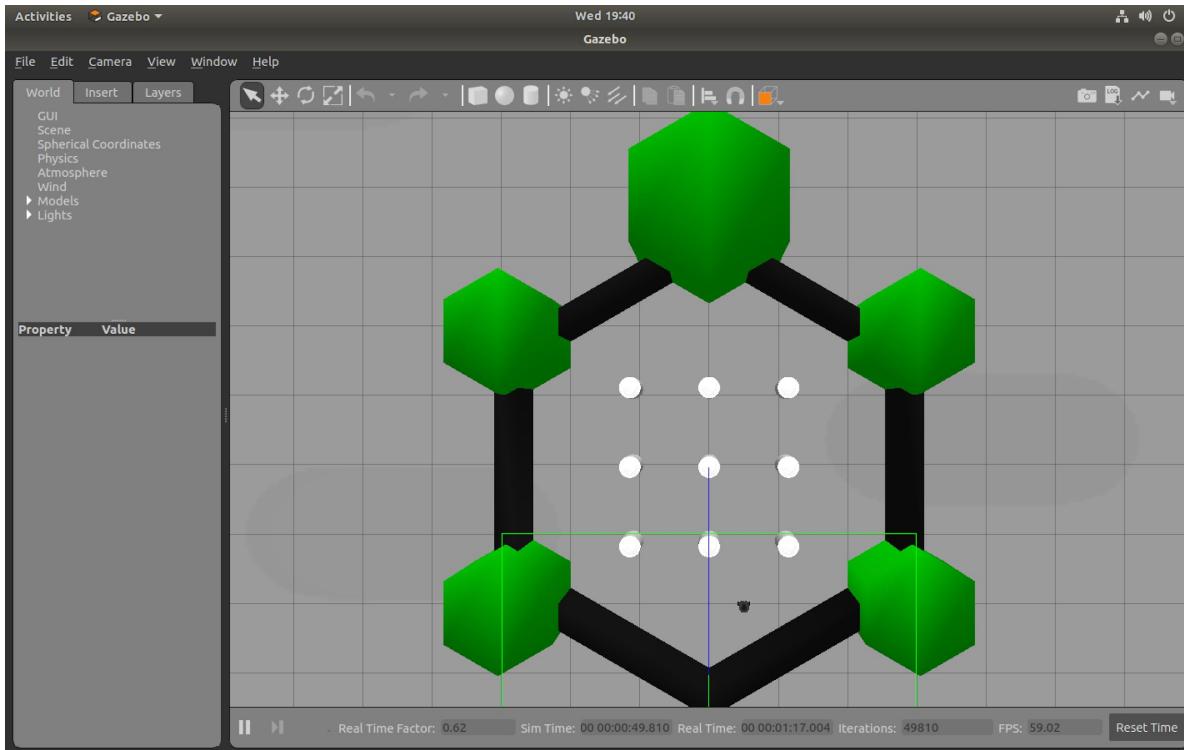
Arg xml is <arg default="\$(env TURTLEBOT3\_MODEL)" doc="model type [burger, waffle, waffle\_pi]" name="model"/>
The traceback for the exception was written to the log file

```
$ export TURTLEBOT3_MODEL=burger
$ roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch
```

# Kill gazebo | ROS

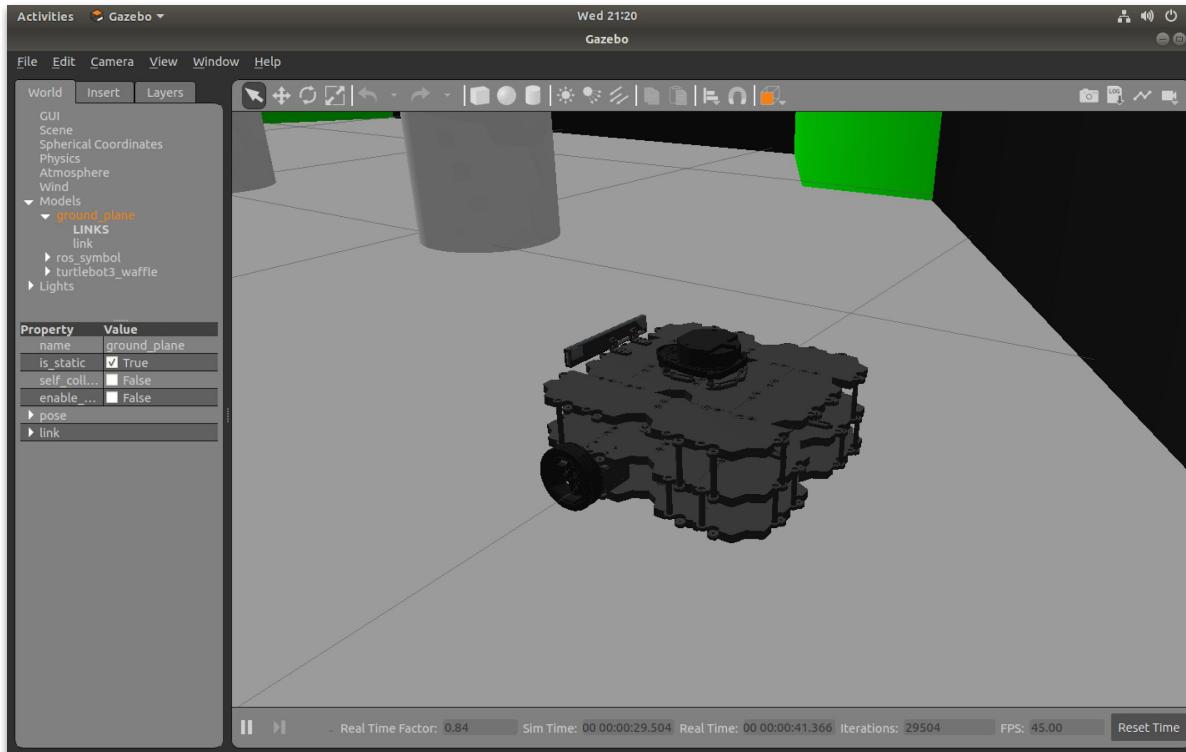
```
$ kill -9 `ps ax | grep gazebo | awk '{ print $1 }'`  
-bash: kill: (6607) - No such process  
$ kill -9 `ps ax | grep ros | awk '{ print $1 }'`  
-bash: kill: (6632) - No such process  
$ # PRO-TIP: Create a command for this!
```

# Turtlebot 3 burger in Gazebo



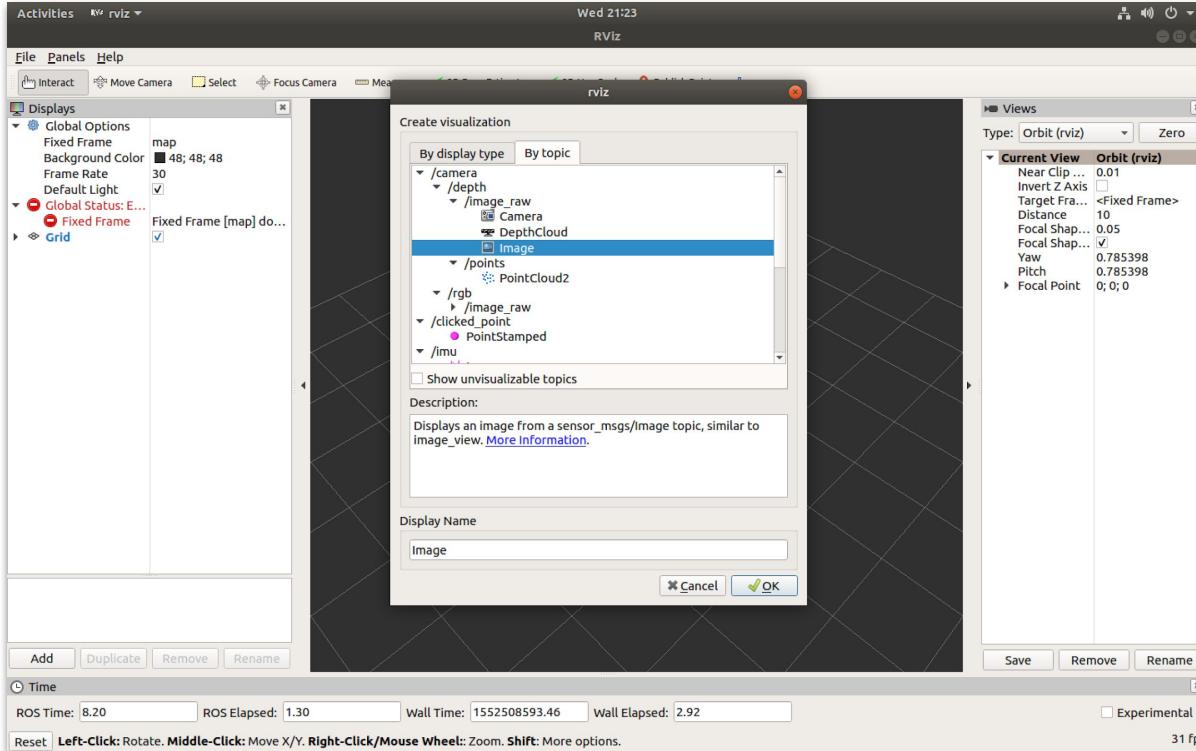
```
$ roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

# Turtlebot 3 waffle in Gazebo



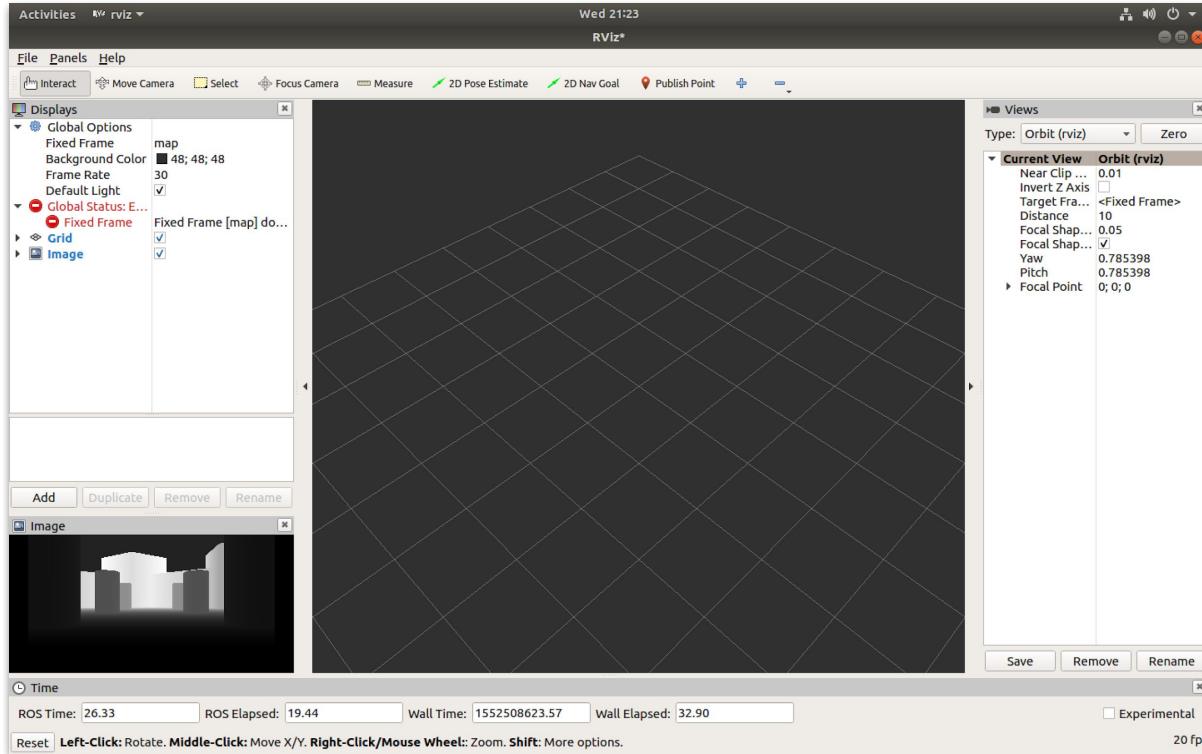
```
$ export TURTLEBOT3_MODEL=waffle; roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

# Turtlebot 3 waffle camera in RVIZ



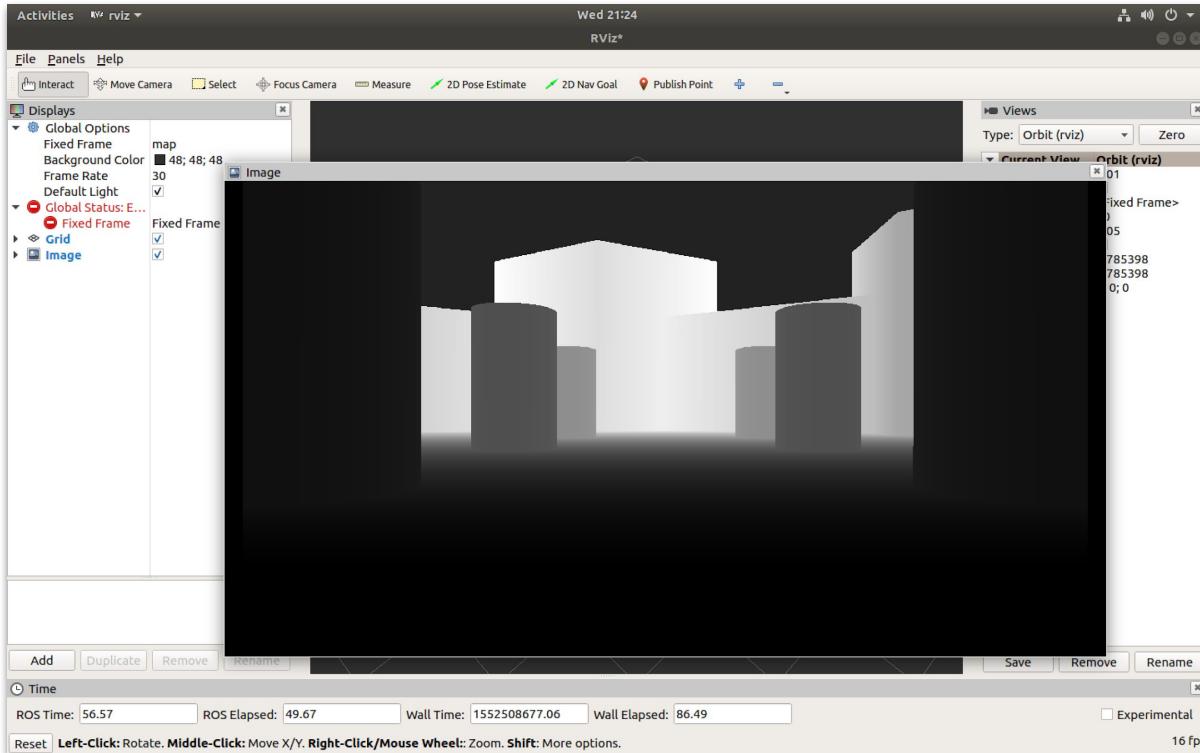
```
$ export TURTLEBOT3_MODEL=waffle; roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

# Turtlebot 3 waffle camera in RVIZ



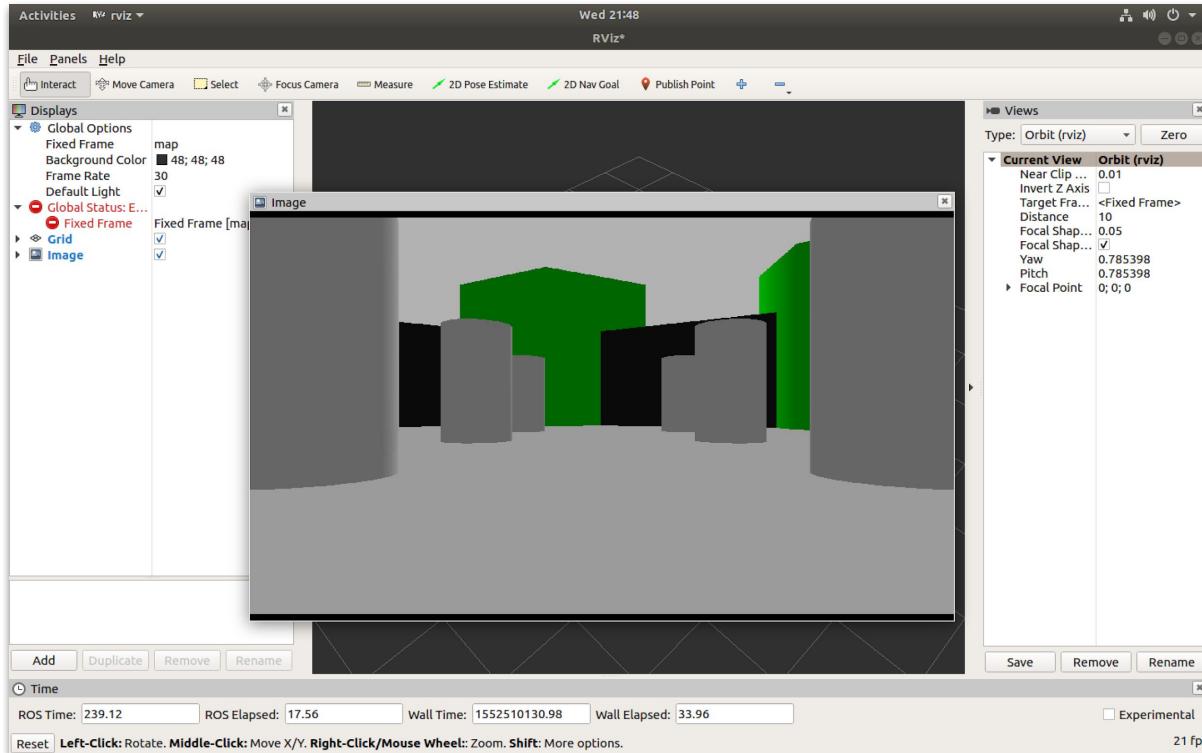
```
$ export TURTLEBOT3_MODEL=waffle; roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

# Turtlebot 3 waffle camera in RVIZ



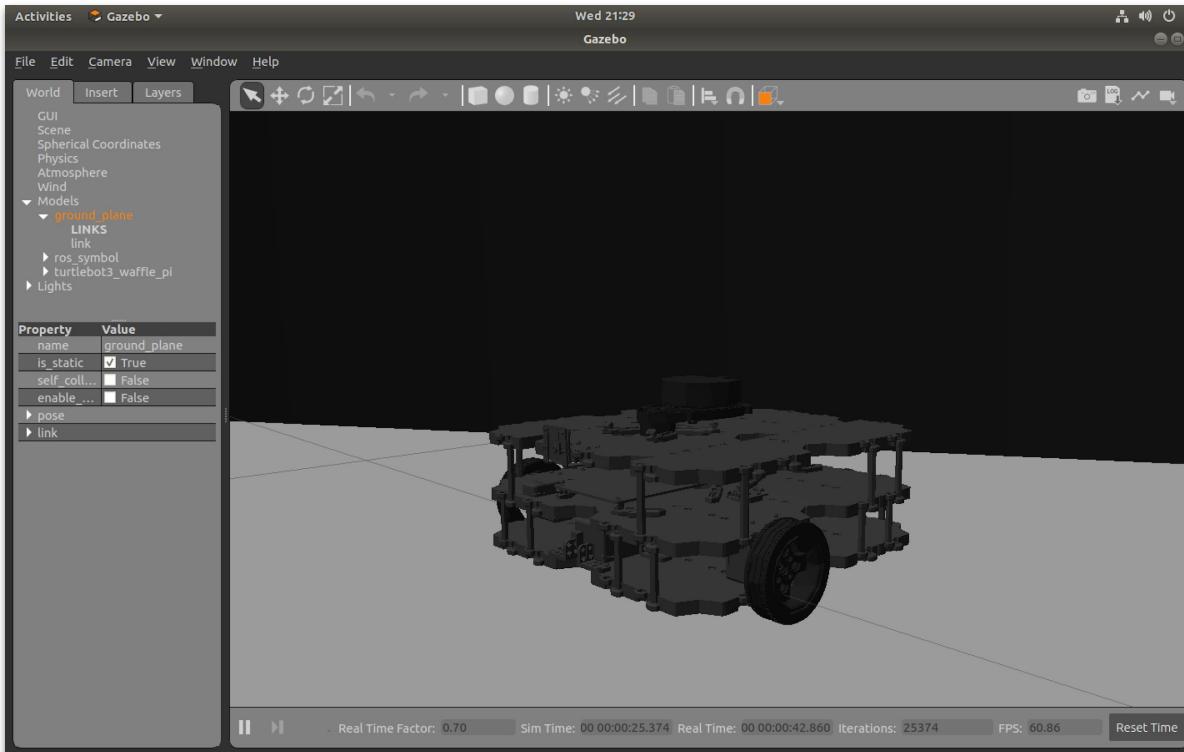
```
$ export TURTLEBOT3_MODEL=waffle; roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

# Turtlebot 3 waffle camera in RVIZ



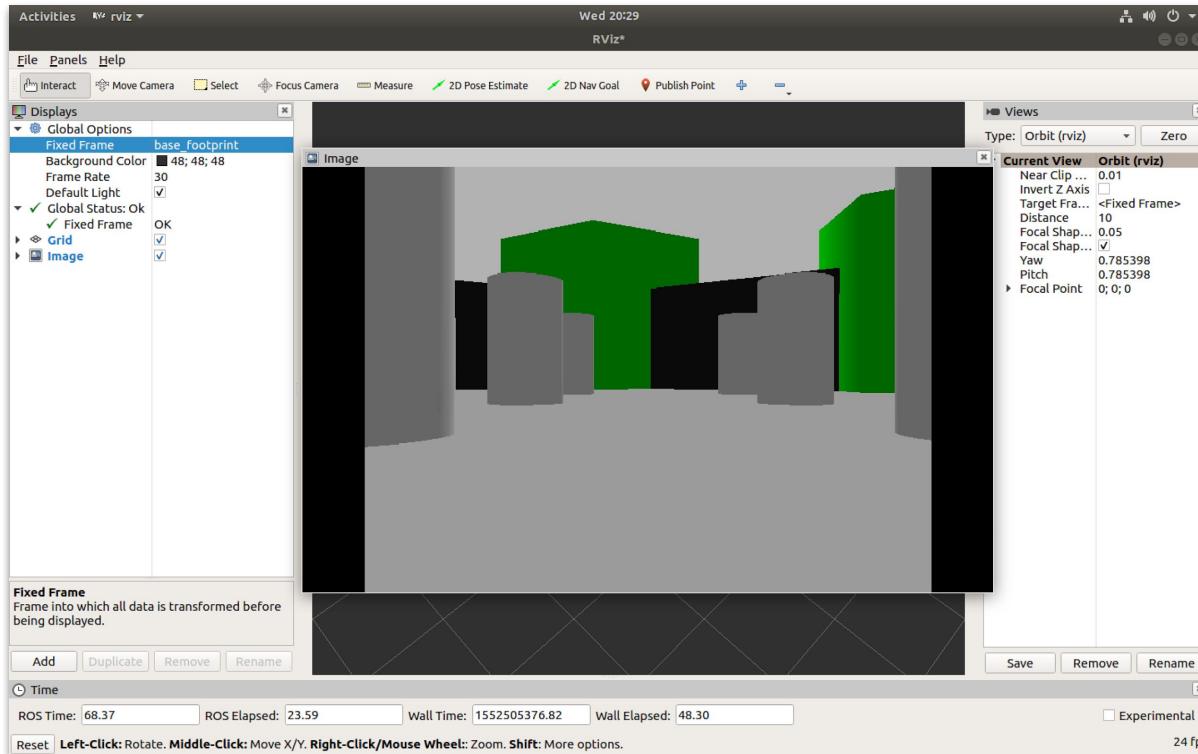
```
$ export TURTLEBOT3_MODEL=waffle; roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

# Turtlebot 3 waffle\_pi in Gazebo



```
$ export TURTLEBOT3_MODEL=waffle_pi; roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

# Turtlebot 3 waffle\_pi camera in RVIZ



```
$ export TURTLEBOT3_MODEL=waffle_pi; roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

# Recording and playing back data

The screenshot shows a web browser displaying a ROS.org tutorial page. The page has a dark blue header with the ROS.org logo, navigation links for Documentation, Browse Software, News, and Download, and a search bar. The main content area has a light blue background. It features a title 'ROS/ Tutorials/ Recording and playing back data', a note about asking questions on answers.ros.org, and a 'Recording and playing back data' section with sub-sections for recording data, examining and playing the bag file, recording a subset of the data, and the limitations of rosbag record/play. On the right side, there are sidebar menus for Wiki, Page, Info, Attachments, and User, along with a 'More Actions' dropdown and a 'Login' link.

**ROS/ Tutorials/ Recording and playing back data**

Please ask about problems and questions regarding this tutorial on [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.

## Recording and playing back data

**Description:** This tutorial will teach you how to record data from a running ROS system into a .bag file, and then to play back the data to produce similar behavior in a running system.

**Keywords:** data, rosbag, record, play, info, bag

**Tutorial Level:** BEGINNER

**Next Tutorial:** [Getting started with rosrun](#)

**Contents**

- 1. Recording data (creating a bag file)
  - 1. Recording all published topics
  - 2. Examining and playing the bag file
  - 3. Recording a subset of the data
  - 4. The limitations of rosbag record/play

### 1. Recording data (creating a bag file)

This section of the tutorial will instruct you how to record topic data from a running ROS system. The topic data will be accumulated in a bag file.

First, execute the following commands in separate terminals:

Terminal 1:

[SOURCE]

<http://wiki.ros.org/ROS/Tutorials/Recording%20and%20playing%20back%20data>

# The limitations of rosbag record/play

In the previous section **you may have noted that the turtle's path may not have exactly mapped to the original keyboard input** - the rough shape should have been the same, but **the turtle may not have exactly tracked the same path**. The reason for this is that the path tracked by **turtlesim** is very **sensitive to small changes in timing in the system**, and **rosbag is limited in its ability to exactly duplicate the behavior of a running system** in terms of when messages are recorded and processed by rosrecord, and when messages are produced and processed when using rospaly. For nodes like turtlesim, where minor timing changes in when command messages are processed can subtly alter behavior, the user should not expect perfectly mimicked behavior.

[[SOURCE](#)]

<http://wiki.ros.org/ROS/Tutorials/Recording%20and%20playing%20back%20data>

# Getting started with roswtf

The screenshot shows the ROS.org website with the following details:

- Header:** ROS.org, About | Support | Discussion Forum | Service Status | Q&A answers.ros.org, Search: [input] Submit.
- Navigation Bar:** Documentation, Browse Software, News, Download.
- Page Title:** ROS/ Tutorials/ Getting started with roswtf
- Text Box (Alert):** Please ask about problems and questions regarding this tutorial on [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.
- Section Header:** Getting started with roswtf
- Description:** Basic introduction to the [roswtf](#) tool.
- Keywords:** roswtf
- Tutorial Level:** BEGINNER
- Next Tutorial:** [Navigating the wiki](#)
- Contents:**
  1. [Checking your installation](#)
  2. [Trying it online](#)
  3. [Errors](#)
- Text:** Before you start this tutorial, please make sure your `roscore` is NOT running.  
On Linux, you can check if `roscore` is still running or not by something like this (if you see a line like this that includes `rosmaster`, which starts as part of `roscore`, `roscore` is running):  

```
$ ps -ef | grep -i rosmaster
00:00:00 /usr/bin/python /opt/ros/kinetic/bin/rosmaster
```
- Section 1:** 1. Checking your installation  

`roswtf` examines your system to try and find problems. Let's try it out:

[SOURCE]

<http://wiki.ros.org/ROS/Tutorials/Getting%20started%20with%20roswtf>

# Understanding ROS Services and Parameters

 ROS.org

About | Support | Discussion Forum | Service Status | Q&A answers.ros.org

Search:  Submit

Documentation    Browse Software    News    Download

**ROS/ Tutorials/ UnderstandingServicesParams**

**Note:** This tutorial assumes that you have completed the previous tutorials: [understanding ROS topics](#).

Please ask about problems and questions regarding this tutorial on [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.

## Understanding ROS Services and Parameters

**Description:** This tutorial introduces ROS services, and parameters as well as using the `rosservice` and `rosparam` commandline tools.

**Tutorial Level:** BEGINNER

**Next Tutorial:** [Using rqt\\_console and roslaunch](#)

**Contents**

- 1. ROS Services
- 2. Using rosservice
  - 1. rosservice list
  - 2. rosservice type
  - 3. rosservice call
- 3. Using rosparam
  - 1. rosparam list
  - 2. rosparam set and rosparam get
  - 3. rosparam dump and rosparam load

Assuming your `turtlesim_node` is still running from the last tutorial, let's look at what services the turtlesim provides:

### 1. ROS Services

Services are another way that nodes can communicate with each other. Services allow nodes to send a **request** and

[SOURCE]

<http://wiki.ros.org/ROS/Tutorials/UnderstandingServicesParams>

# Using rqt\_console and roslaunch

[About](#) | [Support](#) | [Discussion Forum](#) | [Service Status](#) | [Q&A answers.ros.org](#)

Search:

[Documentation](#)    [Browse Software](#)    [News](#)    [Download](#)

[ROS/Tutorials/ UsingRqtconsoleRoslaunch](#)

**Note:** This tutorial assumes that you have completed the previous tutorials: [Understanding ROS services and parameters](#).

Please ask about problems and questions regarding this tutorial on [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.

## Using rqt\_console and roslaunch

**Description:** This tutorial introduces ROS using `rqt_console` and `rqt_logger_level` for debugging and `roslaunch` for starting many nodes at once. If you use ROS Fuerte or earlier distros where `rqt` isn't fully available, please see this page with [this page](#) that uses old `rx` based tools.

**Tutorial Level:** BEGINNER

**Next Tutorial:** [Using rosed](#)

**Contents**

- 1. Prerequisites rqt and turtlesim package
- 2. Using rqt\_console and rqt\_logger\_level
  - 1. Quick Note about logger levels
  - 2. Using rosconsole
- 3. The Launch File
- 4. The Launch File Explained
- 5. roslaunching

### 1. Prerequisites rqt and turtlesim package

The tutorial uses both the `rqt` and `turtlesim` packages. To do this tutorial, please install both packages, if you have not yet done so.

**Wiki**

[Distributions](#)  
[ROS/Installation](#)  
[ROS/Tutorials](#)  
[RecentChanges](#)  
[UsingRqtconsoleRoslaunch](#)

**Page**

[Immutable Page](#)  
[Info](#)  
[Attachments](#)  
More Actions:

**User**

[Login](#)

[SOURCE]

<http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch>

# Creating a ROS msg and srv

The screenshot shows the ROS.org website with the following details:

- Header:** ROS.org logo, navigation links for About, Support, Discussion Forum, Service Status, Q&A answers.ros.org, and a search bar.
- Menu Bar:** Documentation, Browse Software, News, Download.
- Page Title:** ROS/ Tutorials/ CreatingMsgAndSrv
- Note:** A note states: "Note: This tutorial assumes that you have completed the previous tutorials: [using rosed](#)."
- Tip:** A tip message: "💡 Please ask about problems and questions regarding this tutorial on [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."
- Section Header:** Creating a ROS msg and srv
- Description:** This tutorial covers how to create and build msg and srv files as well as the `rosmsg`, `rossrv` and `roscp` commandline tools.
- Tutorial Level:** BEGINNER
- Next Tutorial:** Writing a simple publisher and subscriber ([python](#)) ([c++](#))
- Tags:** catkin, rosbuild
- Contents:**
  - 1. Introduction to msg and srv
  - 2. Using msg
    - 1. Creating a msg
  - 3. Using rosmsg
  - 4. Using srv
    - 1. Creating a srv
    - 2. Using rossrv
  - 5. Common step for msg and srv
  - 6. Getting Help
  - 7. Review
  - 8. Next Tutorial
- Sidebar:** Wiki (Distributions, ROS/Installation, ROS/Tutorials, RecentChanges, **CreatingMsgAndSrv**), Page (Immutable Page, Info, Attachments, More Actions: dropdown), User (Login).

[SOURCE]

<http://wiki.ros.org/ROS/Tutorials/CreatingMsgAndSrv>

# PyStyleGuide

 ROS.org

About | Support | Discussion Forum | Service Status | Q&A answers.ros.org

Search:  Submit

Documentation    Browse Software    News    Download

## PyStyleGuide

### PyStyleGuide

**Contents**

- 1. PyStyleGuide
  - 1. Coding Style
  - 2. The Loader
  - 3. Package/Module Names (`__init__.py` files)
  - 4. Node Files
  - 5. Python Features/Version

This page defines a style guide to be followed in writing Python code for ROS. This guide applies to all ROS code, both core and non-core.

For C++, see the [C++ Style Guide](#) and for Javascript, see the [ROS JavaScript Style Guide](#)

## 1. Coding Style

Python code should follow [PEP 8](#). PEP 8 is not a strict style guide and values readability over consistency, so just try to be smart. A quick summary of PEP 8 is:

- `package_name`
- `ClassName`
- `method_name`
- `field_name`
- `_private_something`
- `self.__really_private_field`
- `_global`
- 4 space indentation

**Wiki**

Distributions  
ROS/Installation  
ROS/Tutorials  
RecentChanges  
**PyStyleGuide**

---

**Page**

Immutable Page  
Info  
Attachments  
More Actions:

---

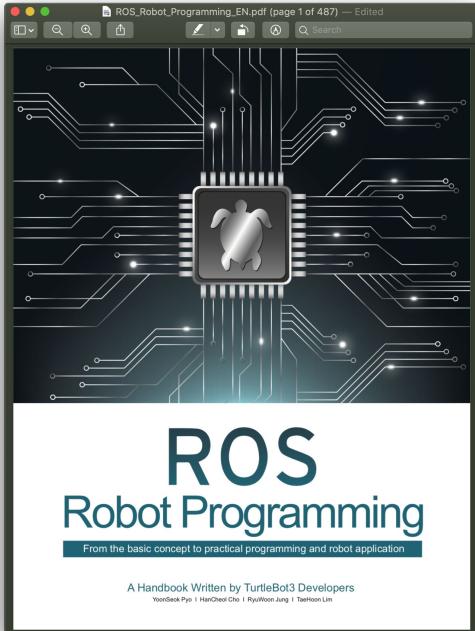
**User**

Login

[SOURCE]

<http://wiki.ros.org/PyStyleGuide>

# ROS Robot Programming

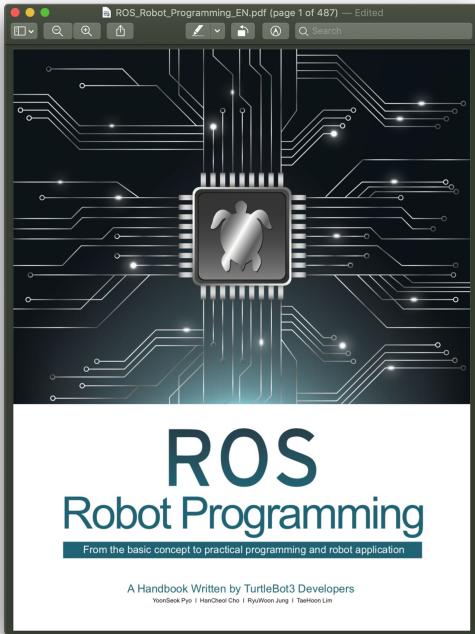


College students and graduate students who want to learn robot programming based on ROS (Robot Operating System) and also for professional researchers and engineers who work on robot development or software programming. We have tried to offer detailed information we learned while working on [TurtleBot3](#) and [OpenManipulator](#). We hope this book will be the **complete handbook for beginners in ROS** and more people will contribute to the ever-growing community of open robotics.

[SOURCE]

[http://wiki.ros.org/Books/ROS\\_Robot\\_Programming\\_English](http://wiki.ros.org/Books/ROS_Robot_Programming_English)

# ROS Robot Programming



From the basic concept to practical robot application programming

- ROS Kinetic Kame: **Basic concept, instructions and tools**
- How to use **sensor** and actuator packages on ROS
- Embedded board for ROS: OpenCR1.0
- SLAM & Navigation with TurtleBot3
- How to program a delivery robot using ROS Java
- OpenManipulator simulation using Movelt! and **Gazebo**

[SOURCE]

[http://wiki.ros.org/Books/ROS\\_Robot\\_Programming\\_English](http://wiki.ros.org/Books/ROS_Robot_Programming_English)

# Why turtles? (just for info)

In addition, there is one more rule. Each version has an illustration in the form of a poster and a turtle icon, as shown in Figure 2-8. These turtle icons are also used in the official simulation tutorial called ‘turtlesim’. The turtle symbol for ROS was stemmed from the educational programming language Logo<sup>16</sup> from MIT’s AI Lab<sup>17</sup> in the 1960s. More than 50 years ago in 1969, a turtle robot was developed using Logo, which was able to actually move on the floor and draw pictures according to the instructions given by the computer. Based on this robot, a program ‘turtlesim’ was developed and the actual robot was later also called TurtleBot<sup>18</sup>.

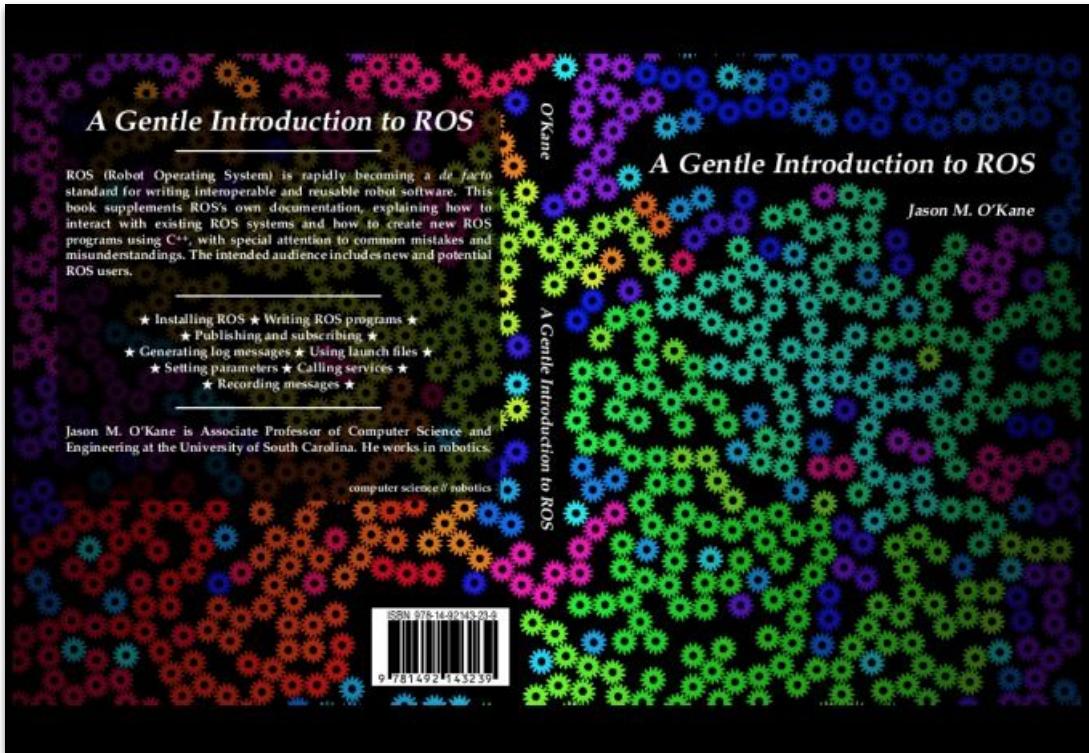


FIGURE 2-8 Turtle icons for each version of ROS

[SOURCE]

[http://wiki.ros.org/Books/ROS\\_Robot\\_Programming\\_English](http://wiki.ros.org/Books/ROS_Robot_Programming_English)

# A Gentle Introduction to ROS



ROS (Robot Operating System) is rapidly becoming a *de facto* standard for writing interoperable and reusable robot software. This book supplements ROS's own documentation, explaining how to interact with existing ROS systems and how to create new ROS programs using C++, with special attention to common mistakes and misunderstandings. The intended audience includes new and potential ROS users.

166 pages

ISBN 978-14-92143-23-9

Printed copies are available from these vendors.

- English version: [Amazon.com](https://www.amazon.com/gp/product/9781492145239)
- Chinese translation: [JD.com](https://www.jd.com/gentle-introduction-to-ros.html)  
[Dangdang.com](https://www.dangdang.com/gentle-introduction-to-ros-9781492145239.html)

Electronic copies are free.

[[SOURCE](#)]

<https://cse.sc.edu/~jokane/agitr/>

