

# ROS Reference

Matt Kontz

Version 1.1 from Dec 17, 2018

## 1 Summary

This document is brief reference covering ROS terminology and how to use ROS.

## 2 About ROS

From [www.ros.org](http://www.ros.org) ...

*The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.*

- Somewhere between middle-ware and framework for robotics application
- Provides separation and communication tools
- Provides plug & play libraries

### 2.1 ROS Goals:

- Provide a standard for robotics application to facilitate reuse, avoid re-inventing wheel.
- Enable modular software design
- Use on any robot
- Language agnostic: C++, Python or other languages (C++ & Python are standard)
- Open source community

### 3 Install ROS

Setup computer or VM with an OS (e.g. 16.04 - Xenial)) that matches the desired version of ROS (e.g. Kinitic Kane)

Find installation instruction on [www.ros.org](http://www.ros.org) for desired version of ROS and operating system (e.g. Kinitic and Ubuntu 16.04 (Xenial))

Verify the *source /opt/ros/kinetic/setup.bash* has been added to *.bashrc*.

### 4 Commands

- Add new package:  
    >> sudo apt-get install ros-kinetic-*package\_i*
- launch ROS master:  
    >> roscore
- Make catkins workspace:
  - create folder:  
    >> mkdir <folder\_name>
  - change directory into folder:  
    >> cd <folder\_name>
  - create src folder:  
    >> cd src
  - initialize workspace:  
    >> catkin\_make
  - [Optional] add setup.bash to .bashrc to avoid having to load setup.bash for this workspace each time  
    >> source /<path>/<catkin\_workspace\_name>/devel/setup.bash

### 5 ROS Tutorials

- [wiki.ros.org/ROS/Tutorials](http://wiki.ros.org/ROS/Tutorials)
- [/www.udemy.com](http://www.udemy.com)

### 6 Terminology

- **nodes** A node is a process that performs computation. Nodes are combined together into a graph and communicate with one another using streaming topics, RPC services, and the Parameter Server. ([wiki.ros.org/Nodes](http://wiki.ros.org/Nodes))

- **topics** Topics are named buses over which nodes exchange messages. Topics have anonymous publish/subscribe semantics, which decouples the production of information from its consumption. In general, nodes are not aware of who they are communicating with. Instead, nodes that are interested in data subscribe to the relevant topic; nodes that generate data publish to the relevant topic. There can be multiple publishers and subscribers to a topic. ([wiki.ros.org/Topics](http://wiki.ros.org/Topics))
- **catkin** Low-level build system macros and infrastructure for ROS. ([wiki.ros.org/catkin](http://wiki.ros.org/catkin))
- **catkin workspace** catkin packages can be built as a standalone project, in the same way that normal cmake projects can be built, but catkin also provides the concept of workspaces, where you can build multiple, interdependent packages together all at once. ([wiki.ros.org/catkin/workspaces](http://wiki.ros.org/catkin/workspaces))
- **package** Software in ROS is organized in packages. A package might contain ROS nodes, a ROS-independent library, a dataset, configuration files, a third-party piece of software, or anything else that logically constitutes a useful module. ([wiki.ros.org/Packages](http://wiki.ros.org/Packages))

## Revision History

Revision	Date	Author(s)	Description
1.0	Dec 21, 2018	Matt Kontz	Created initial version ROS reference.