

💡 Please ask about problems and questions regarding this tutorial on [answers.ros.org](http://answers.ros.org) (<http://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.

# Managing System dependencies

**Description:** This explains how to use rosdep (/rosdep) to install system dependencies.

**Tutorial Level:** INTERMEDIATE

**Next Tutorial:** Roslaunch tips for large projects (/ROS/Tutorials/Roslaunch%20tips%20for%20larger%20projects)

## Sommaire

1. System Dependencies
  1. rosdep
  2. rosdistro/rosdep

[indigo](#)[kinetic](#)[lunar](#)[melodic](#)

Show EOL distros: ☐

## 1. System Dependencies

ROS packages sometimes require external libraries and tools that must be provided by the operating system. These required libraries and tools are commonly referred to as *system dependencies*. In some cases these *system dependencies* are not installed by default. ROS provides a simple tool, rosdep, that is used to download and install *system dependencies*.

ROS packages must declare that they need these *system dependencies* in the package manifest. Let's look at the manifest for the turtlesim package:

```
$ roscd turtlesim
```

Then,

```
$ cat package.xml
```

```
<package>

...
...
  <build_depend>message_generation</build_depend>
  <build_depend>libqt4-dev</build_depend>
  <build_depend>qt4-qmake</build_depend>
  <build_depend>roscpp</build_depend>
  <build_depend>roscpp_serialization</build_depend>
  <build_depend>roslib</build_depend>
  <build_depend>rostime</build_depend>
  <build_depend>std_msgs</build_depend>
  <build_depend>std_srvs</build_depend>
</package>
```

As you can see turtlesim (/turtlesim) needs those libraries and packages.

## 1.1 rosdep

rosdep is a tool you can use to install system dependencies required by ROS packages.

Usage:

```
rosdep install [package]
```

Download and install the system dependencies for turtlesim:

```
$ rosdep install turtlesim
```

If you've been following along with the tutorials, it's likely that this is the first time you've used rosdep. When you run this command, you'll get an error message:

```
ERROR: your rosdep installation has not been initialized yet. Please run:

    sudo rosdep init
    rosdep update
```

Just run those two commands and then try to install turtlesim's dependencies again.

If you installed using binaries you will see:

```
All required rosdeps installed successfully
```

Otherwise you will see the output of installing the dependencies of turtlesim:

```
#!/usr/bin/bash



set -o errexit
set -o verbose

if [ ! -f /opt/ros/lib/libboost_date_time-gcc42-mt*-1_37.a ] ; then
  mkdir -p ~/ros/ros-deps
  cd ~/ros/ros-deps
  wget --tries=10 http://pr.willowgarage.com/downloads/boost_1_37_0.tar.gz
  z
  tar xzf boost_1_37_0.tar.gz
  cd boost_1_37_0
  ./configure --prefix=/opt/ros
  make
  sudo make install
fi

if [ ! -f /opt/ros/lib/liblog4cxx.so.10 ] ; then
  mkdir -p ~/ros/ros-deps
  cd ~/ros/ros-deps
  wget --tries=10 http://pr.willowgarage.com/downloads/apache-log4cxx-0.10.0-wg_patched.tar.gz
  tar xzf apache-log4cxx-0.10.0-wg_patched.tar.gz
  cd apache-log4cxx-0.10.0
  ./configure --prefix=/opt/ros
  make
  sudo make install
fi
```

rosdep runs the bash script above and exits when complete.

## 1.2 rosdistro/rosdep

While  rosdep (<http://wiki.ros.org/ROS/Tutorials/rosdep#rosdep>) is the client tool, the reference is provided by rosdep rules, stored online in  ros/roscdistro/rosdep on github (<https://github.com/ros/roscdistro/tree/master/rosdep>).


When doing

```
$ rosdep update
```

rosdep actually retrieves the rules from the roscdistro github repository.

As of version 0.14.0 rosdep update will only fetch ROS package names for non-EOL ROS distributions. If you are still using an EOL ROS distribution (/Distributions) (which you probably shouldn't) you can pass the argument `--include-eol-distros` to also fetch the ROS package names of those.

These rules are used when a dependency is listed that doesn't match the name of a ROS package built on the buildfarm. Then rosdep checks if there exists a rule to resolve it for the proper platform and package manager you are using.

When creating a new package, you might need to declare new system dependencies to the  rosdep rules (<https://github.com/ros/rosdistro/tree/master/rosdep>) if they are not there yet. Just edit the file, add the dependency needed (following a strict alphabetical order and a similar structure as the other dependencies already registered) and send a pull request.

After that pull request has been merged, you need to run :

```
$ rosdep update
```

and now that dependency will be resolved by rosdep.

You can test it with :

```
$ rosdep resolve my_dependency_name
```

The output should be something like :

```
#apt
my-dependency-name
```

where the first line is the package manager chosen for installing this dependency, and the second line is the actual name for that dependency on your current platform.

Except where otherwise noted, the

ROS wiki is licensed under the

Wiki: ROS/Tutorials/rosdep (dernière édition le 2019-01-27 12:57:47 par IsaacSaito (/IsaacSaito))

Creative Commons Attribution 3.0

(<http://creativecommons.org/licenses/by/3.0/>)

Brought to you by:  Open Source Robotics Foundation

(<http://www.osrfoundation.org>)