💡 It is appreciated that problems/questions regarding this tutorial are asked on answers.ros.org (http://answers.ros.org). Don't forget to include in your question the link to this page, versions of your OS & ROS, and also add appropriate tags.

1. Installing ROS Indigo on Debian Wheezy

Description: These are instructions to install Indigo on Debian Wheezy.

Keywords: Debian, Setup, Indigo

Tutorial Level: BEGINNER

Contents

Introduction

This tutorial explains how to install ROS Indigo from source on Debian Wheezy. The instructions roughly follow the source installation on Raspbian OS (which is also Debian-based) (/indigo/Installation/Source). However, a few dependencies need to be accounted spcifically for Debian Wheezy.

1. Prerequisites

These instructions assume that Debian (wheezy) is being used. The download page for current images of Debian Wheezy is https://www.debian.org/releases/stable/debian-installer/ (https://www.debian.org/releases/stable/debian-installer/).

1.1 Setup ROS Repositories

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu wheezy main" > /etc/apt/sources.list.d/ros-latest.list'
$ wget https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -O - | sudo apt-key add -
```

1.2 Enable Wheezy backports repository

Backports are packages from testing or unstable versions od Debian ported back to the stable version to allow access to cutting-edge versions of software on an otherwise stable system.

```
$ sudo sh -c 'echo "deb http://http.debian.net/debian wheezy-backports main" > /etc/apt/sources.list.d/backports.list'
```

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

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

1.3 Install bootstrap dependencies

```
$ sudo apt-get install python-setuptools
$ sudo easy_install pip
$ sudo pip install -U rosdep rosinstall_generator wstool rosinstall
```

1.4 Initializing rosdep

```
$ sudo rosdep init
$ rosdep update
```

2. Installation

Now we will download and build ROS Indigo.

2.1 Create a catkin Workspace

In order to build the core packages, you will need a catkin workspace. Create one now:

```
$ mkdir ~/ros_catkin_ws
$ cd ~/ros_catkin_ws
```

Next we will want to fetch the core packages so we can build them. We will use wstool for this. Select the wstool command for the particular variant you want to install:

ROS-Comm: (recommended) ROS package, build, and communication libraries. No GUI tools.

```
$ rosinstall_generator ros_comm --rosdistro indigo --deps --wet-only --exclude roslisp --tar > indigo-ros_comm-wet.rosinstall
$ wstool init -j8 src indigo-ros_comm-wet.rosinstall
```

Desktop: ROS, rqt (/rqt), rviz (/rviz), and robot-generic libraries

```
$ rosinstall_generator desktop --rosdistro indigo --deps --wet-only --exclude roslisp --tar > indigo-desktop-wet.rosinstall
$ wstool init -j8 src indigo-desktop-wet.rosinstall
```

This will add all of the catkin or wet packages in the given variant and then fetch the sources into the ~/ros_catkin_ws/src directory. The command will take a few minutes to download all of the core ROS packages into the src folder. The -j8 option downloads 8 packages in parallel.

Note: The roslisp package is excluded in these generators because it is not tested. If it works for you, please edit this tutorial.

So far, only these two variants have been tested on Debian Wheezy in Indigo; however, more are defined in REP 131 (http://ros.org/reps/rep-0131.html#variants) such as robot, perception, etc. Just change the package path to the one you want, e.g., for robot do:

```
$ rosinstall_generator robot --rosdistro indigo --deps --wet-only --tar > indigo-robot-wet.rosinstall
$ wstool init -j8 src indigo-robot-wet.rosinstall
```

Please feel free to update these instructions as you test more variants.

If wstool init fails or is interrupted, you can resume the download by running:

```
wstool update -j 4 -t src
```

2.2 Resolve Dependencies

Before you can build your catkin workspace you need to make sure that you have all the required dependencies. We use the rosdep tool for this, however, first, a couple of dependencies not available in the repositories must be manually built.

2.2.1 Dependencies not available in the Debian Wheezy branch

At the time this article was written (01/2015) several packages: libconsole-bridge-dev, liburdfdom-headers-dev, liburdfdom-dev, liblz4-dev, and collada-dom-dev are not in the Debian stable branch, and libopencv-dev and OGRE are old versions. The following are needed for each variant:

- Needed for **Ros_Comm**: libconsole-bridge-dev and liblz4-dev
- Needed for **Desktop**: libconsole-bridge-dev, liblz4-dev, liburdfdom-headers-dev, liburdfdom-dev, collada-dom-dev and OGRE and libopencv-dev

These packages can be built from source in a new directory (Also install checkinstall and cmake):

```
$ mkdir ~/ros_catkin_ws/external_src
$ sudo apt-get install checkinstall cmake
```

libconsole-bridge-dev: Install with the following:

```
$ cd ~/ros_catkin_ws/external_src
$ sudo apt-get install libboost-system-dev libboost-thread-dev
$ git clone https://github.com/ros/console_bridge.git
$ cd console_bridge
$ cmake .
$ sudo checkinstall make install
```

When check-install asks for any changes, the name (2) needs to change from "console-bridge" to "libconsole-bridge-dev" otherwise rosdep install wont find it. You can also skip generating documentation.

liburdfdom-headers-dev: Install with the following:

```
$ cd ~/ros_catkin_ws/external_src
$ git clone https://github.com/ros/urdfdom_headers.git
$ cd urdfdom_headers
$ cmake .
$ sudo checkinstall make install
```

When check-install asks for any changes, the name (2) needs to change from "urdfdom-headers" to "liburdfdom-headers-dev" otherwise rosdep install wont find it.

liburdfdom-dev: Install with the following:

```
$ cd ~/ros_catkin_ws/external_src
$ sudo apt-get install libboost-test-dev libtinyxml-dev
$ git clone https://github.com/ros/urdfdom.git
$ cd urdfdom
$ cmake .
$ sudo checkinstall make install
```

When check-install asks for any changes, the name (2) needs to change from "urdfdom" to "liburdfdom-dev" otherwise rosdep install wont find it.

collada-dom-dev: Install with the following (Note: You will also need to patch collada_urdf as described here (https://github.com/ros/robot_model/issues/12)):

```
$ cd ~/ros_catkin_ws/external_src
$ sudo apt-get install libboost-filesystem-dev libxml2-dev
$ wget http://downloads.sourceforge.net/project/collada-dom/Collada%20DOM/Collada%20DOM%202.4/collada-dom-
2.4.0.tgz
$ tar -xzf collada-dom-2.4.0.tgz
$ cd collada-dom-2.4.0
$ cmake .
$ sudo checkinstall make install
```

When check-install asks for any changes, the name (2) needs to change from "collada-dom" to "collada-dom-dev" otherwise rosdep install wont find it.

libopencv-dev: Install with the following (be prepared it's gonna take more than an hour; based on http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html (http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html)):

```
$ cd ~/ros_catkin_ws/external_src
$ git clone https://github.com/Itseez/opencv.git
$ mkdir opencv/release
$ cd opencv/release
$ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..
$ make
$ sudo checkinstall make install
```

When check-install asks for any changes, the name (2) needs to change from "release" to "libopencv-dev" otherwise rosdep install wont find it.

If you have some problems compiling the newest OpenCV, you can try to compile a release version by switching to the 2.4 branch. After git clone, type git checkout 2.4.

Also, if there is an existing installation of OpenCV 2.3 on the system, you may have to first uninstall it to avoid compilation and installation problems.

liblz4-dev: Install using Wheezy backports:

```
$ sudo apt-get -t wheezy-backports install liblz4-dev
```

OGRE: Install newer version through apt-get

```
$ sudo apt-get remove libogre-1.7.*
$ sudo apt-get install libogre-1.8-dev
```

Note: If you don't want to compile Collada but would like to install the desktop variant, use the following generator:

```
$ rosinstall_generator desktop --rosdistro indigo --deps --wet-only --exclude roslisp collada_parser collada_u
rdf --tar > indigo-desktop-wet.rosinstall
```

2.2.2 Resolving Dependencies with rosdep

The remaining dependencies should be resolved by running rosdep:

```
$ cd ~/ros_catkin_ws
$ rosdep install --from-paths src --ignore-src --rosdistro indigo -y -r --os=debian:wheezy
```

This will look at all of the packages in the src directory and find all of the dependencies they have. Then it will recursively install the dependencies.

The --from-paths option indicates we want to install the dependencies for an entire directory of packages, in this case src. The --ignore-src option indicates to rosdep that it shouldn't try to install any ROS packages in the src folder from the package manager, we don't need it to since we are building them ourselves. The --rosdistro option is required because we don't have a ROS environment setup yet, so we have to indicate to rosdep what version of ROS we are building for. Finally, the -y option indicates to rosdep that we don't want to be bothered by too many prompts from the package manager.

After a while rosdep will finish installing system dependencies and you can continue.

Note: Rosdep and apt-get may report that python-rosdep, python-catkin-pkg, python-rospkg, and python-rosdistro failed to install; however, you can ignore this error because they have already been installed with pip.

2.3 Building the catkin Workspace

Once you have completed downloading the packages and have resolved the dependencies, you are ready to build the catkin packages.

Invoke catkin_make_isolated:

```
$ sudo ./src/catkin/bin/catkin_make_isolated --install -DCMAKE_BUILD_TYPE=Release --install-space /opt/ros/indi
go
```

Note: This will install ROS in the equivalent file location to Ubuntu in /opt/ros/indigo however you can modify this as you wish.

Now ROS should be installed! Remember to source the new installation:

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

3. Maintaining a Source Checkout

3.1 Updating the workspace

See the Ubuntu source install instructions (/indigo/Installation/Source) for steps on updating the ros_catkin_ws workspace. The same steps should apply to Debian Wheezy.

3.2 Adding released packages

You may add additional packages to the installed ros workspace that have been released into the ros ecosystem. First, a new rosinstall file must be created including the new packages (Note, this can also be done at the initial install). For example, if we have installed ros_comm, but want to add ros_control and joystick_drivers, the command would be:

```
$ cd ~/ros_catkin_ws
$ rosinstall_generator ros_comm ros_control joystick_drivers --rosdistro indigo --deps --wet-only --exclude ros
lisp --tar > indigo-custom_ros.rosinstall
```

You may keep listing as many ROS packages as you'd like separated by spaces.

Next, update the workspace with wstool:

```
$ wstool merge -t src indigo-custom_ros.rosinstall
$ wstool update -t src
```

After updating the workspace, you may want to run rosdep to install any new dependencies that are required:

```
$ rosdep install --from-paths src --ignore-src --rosdistro indigo -y -r --os=debian:wheezy
```

Finally, now that the workspace is up to date and dependencies are satisfied, rebuild the workspace:

```
$ sudo ./src/catkin/bin/catkin_make_isolated --install -DCMAKE_BUILD_TYPE=Release --install-space /opt/ros/indigo
```

4. References

- http://wiki.ros.org/indigo/Installation/Source (http://wiki.ros.org/indigo/Installation/Source)
- http://wiki.ros.org/hydro/Installation/Debian (http://wiki.ros.org/hydro/Installation/Debian)
- http://wiki.ros.org/ROSberryPi/Setting%20up%20Hydro%20on%20RaspberryPi (http://wiki.ros.org/ROSberryPi/Setting%20up%20Hydro%20on%20RaspberryPi)
- http://wiki.ros.org/ROSberryPi/Installing%20ROS%20Indigo%20on%20Raspberry%20Pi (http://wiki.ros.org/ROSberryPi/Installing%20ROS%20Indigo%20on%20Raspberry%20Pi)

Wiki: indigo/Installation/Debian (last edited 2015-01-30 13:04:44 by Martin Pecka (/Martin%20Pecka))