

# ROS By Example Cheat Sheet for ROS Fuerte

The [ROS By Example](#) book was written for ROS Electric. Since then, ROS Fuerte was released and introduced a number of changes to the launch files and some of the Python code used in the book. This document summarizes the changes required to use the ROS By Example book and code while running ROS Fuerte instead of Electric.

The changes are listed by section as they appear in the book.

## Credits

Many thanks to **Ross Lunan** (member of the Home Brew Robotics Club) for his helpful notes on running the ROS By Example code under ROS Fuerte and Ubuntu 12.04 (Precise).

## Section 5.1: Installing the Prerequisites

To install the prerequisite packages under Fuerte, copy and paste the following command. The "\" character at the end of each line tells Linux to treat the entire string as a single line:

```
$ sudo apt-get install ros-fuerte-turtlebot* ros-fuerte-control \
ros-fuerte-openni-kinect ros-fuerte-laser-drivers ros-fuerte-audio-common \
ros-fuerte-joystick-drivers ros-fuerte-orocos-kinematics-dynamics \
ros-fuerte-dynamixel-motor gstreamer0.10-pocketsphinx python-setuptools
```

## Section 5.2: Installing the rbx\_vol\_1 Stack

### IMPORTANT!!

If you have already installed the ROS By Example stack under Electric, be sure to update it to the latest release as follows:

```
$ roscd rbx_vol_1
$ svn update
$ rosmake -pre-clean
```

## Section 6.1: Installing the Simulator

Instead of this:

```
$ cd ~/ros_workspace
$ sudo apt-get install ros-electric-control
$ svn checkout http://vanadium-ros-pkg.googlecode.com/svn/trunk/arbotix
$ rosmake arbotix
```

do this:

```
$ cd ~/ros_workspace
$ sudo apt-get install ros-fuerte-control
$ svn checkout http://vanadium-ros-pkg.googlecode.com/svn/trunk/arbortix
$ rosmake --pre-clean arbortix
```

**NOTE:** Even if you have already run `rosmake` on the `arbortix` stack under Electric, it is important to run it again now under Fuerte as shown above.

## New Fuerte RViz Config File Names

New Fuerte RViz config files have been added to the ROS By Example stack. So whenever the book uses a config file called `x_electric.vcg`, use `x_fuerte.vcg` instead.

For example, in **Section 6.2**, instead of the command:

```
$ rosrun rviz rviz -d `rospack find rbx1_nav`/sim_electric.vcg
```

use this:

```
$ rosrun rviz rviz -d `rospack find rbx1_nav`/sim_fuerte.vcg
```

**NOTE:** The RViz **View** menu is now called **Panels**.

## New Fuerte Launch File Names

In many cases, you can simply use the newly added Fuerte version of a launch file. For example, in **Section 8.3.2**, instead of

```
$ roslaunch rbx1_bringup kinect.launch
```

you would use:

```
$ roslaunch rbx1_bringup kinect_fuerte.launch
```

## Section 7.4: Calibrating Your Robot's Odometry

Instead of:

```
$ sudo apt-get install ros-electric-orocos-kinematics-dynamics
```

use this:

```
$ sudo apt-get install ros-fuerte-orocos-kinematics-dynamics
```

## Section 7.10: Teleoperating your Robot

Instead of:

```
$ sudo apt-get install ros-electric-joystick-drivers ros-electric-turtlebot-apps
```

use this:

```
$ sudo apt-get install ros-fuerte-joystick-drivers ros-fuerte-turtlebot-apps
```

## Section 8.3.1 Testing move\_base without Obstacles

Remember to use the Fuerte version of the RViz config file:

```
$ rosrn rviz rviz -d `rospack find rbx1_nav`/nav_ekf_fuerte.vcg
```

## Section 8.3.2: Avoiding Obstacles using the TurtleBot's Fake Laser Scan

Instead of:

```
$ roslaunch rbx1_bringup kinect.launch
```

use this:

```
$ roslaunch rbx1_bringup kinect_fuerte.launch
```

## Section 8.4.1: Laser Scanner or Depth Camera?

Instead of:

```
$ sudo apt-get install ros-electric-laser-drivers
```

use this:

```
$ sudo apt-get install ros-fuerte-laser-drivers
```

Replace the reference to `fake_laser.launch` with `fake_laser_fuerte.launch`.

## Section 8.5.2: Using amcl with a Real Robot

Instead of:

```
$ roslaunch rbx1_bringup kinect.launch
```

use this:

```
$ roslaunch rbx1_bringup kinect_fuerte.launch
```

## Section 9.1: Installing PocketSphinx for Speech Recognition

Instead of:

```
$ sudo apt-get install ros-electric-sound-drivers
```

use this:

```
$ sudo apt-get install ros-fuerte-audio-common  
$ sudo apt-get install libasound2
```

## Section 9.5: Installing and Testing Festival Text-to-Speech

Instead of:

```
$ sudo apt-get install ros-electric-sound-drivers
```

use this:

```
$ sudo apt-get install ros-fuerte-audio-common  
$ sudo apt-get install libasound2
```

## Section 10.3.1: Installing the OpenNI Drivers

Instead of:

```
$ sudo apt-get install ros-electric-openni-kinect
```

use this:

```
$ sudo apt-get install ros-fuerte-openni-kinect
```

## Section 10.3.3: Testing your Kinect or Xtion Camera

Use the Fuerte version of the `openni_node` launch file:

```
$ roslaunch rbx1_vision oppenni_node_fuerte.launch
```

**NOTE:** Remember to always use this Fuerte version of the launch file throughout the vision chapter.

## Section 10.3.2: Installing Webcam Drivers

Eric Perko's `uvc_cam` package has been updated for ROS Fuerte so be sure to get the update if you have already installed the package under ROS Electric:

```
$ roscd uvc_cam
$ git pull
$ rosmake --pre-clean
```

You can also try the `camera_umd` driver by installing the following packages:

```
$ sudo apt-get install ros-fuerte-camera-umd ros-fuerte-camera-drivers
```

### Section 10.3.4 Testing your USB Webcam

If you are using a webcam, make sure it can be detected under ROS Fuerte:

```
$ roslaunch rbx1_vision uvc_cam.launch device:=/dev/video0
```

or

```
$ roslaunch rbx1_vision uvc_cam.launch device:=/dev/video1
```

depending on the video device that points to your camera. Note that the `uvc_cam` node publishes the video stream on the topic `/camera/image_raw` by default. The `uvc_cam.launch` launch file remaps this topic to `/camera/rgb/image_color` which is the topic used by the `openni` driver for depth cameras. This way we can use the same code below for either type of camera.

#### Option 2

If the `uvc_cam` driver does not work with your camera, you can try the `camera_umd` package instead with one of the following commands:

```
$ roslaunch rbx1_vision uvc_cam_fuerte.launch device:=/dev/video0
```

or

```
$ roslaunch rbx1_vision uvc_cam_fuerte.launch device:=/dev/video1
```

depending on the camera you want use. Note that the `camera_umd` node publishes the video stream on the topic `/image_raw` by default. The `uvc_cam_fuerte.launch` launch file remaps this topic to `/camera/rgb/image_color` which is the topic used by the `openni` driver for depth cameras. This way we can use the same code below for either type of camera.

### Section 10.4 Installing OpenCV on Ubuntu Linux

Install the Fuerte version of OpenCV:

```
$ sudo apt-get install ros-fuerte-opencv2 ros-fuerte-vision-opencv
```

If you are running *both* Electric and Fuerte on the same machine, you can leave `libopencv2` installed. However, if you have only Fuerte installed, you can remove the older OpenCV library like this:

```
$ sudo apt-get remove libopencv2*
```

If you get prompted to also remove a bunch of ROS Electric packages, then you actually have Electric installed as well and you should abort by answering "n" or typing Ctrl-C.

## Section 10.7 Processing Recorded Video

There is an error in the original text. The command shown as:

```
$ roslaunch rbx1_vision video2ros.launch input:=`rospack find  
rbx1_vision`/videos/hide2.mp4
```

should be a single line using the "\"" character at the first line break below:

```
$ roslaunch rbx1_vision video2ros.launch input:="`rospack find \  
rbx1_vision`/videos/hide2.mp4"
```

## Section 10.9.1 Viewing Skeletons in RViz

Remember to use the Fuerte version of the RViz config file:

```
$ rosrun rviz rviz -d `rospack find rbx1_vision`/skeleton_frames_fuerte.vcg
```

## Section 10.9.2 Accessing Skeleton Frames in your Programs

If you already have the `skeleton_markers` package installed from an early time, be sure to update it now to the latest version:

```
$ roscd skeleton_markers  
$ svn update  
$ rosmake --pre-clean
```

Once updated, use the Fuerte versions of the launch file and RViz config file:

```
$ roslaunch skeleton_markers markers_from_tf_fuerte.launch
```

and

```
$ rosrun rviz rviz -d `rospack find \  
skeleton_markers`/markers_from_tf_fuerte.vcg
```

## Section 10.10.1 The PassThrough Filter

Remember to use the Fuerte versions of the `openni_node` launch file and the RViz config file:

```
$ roslaunch rbx1_vision oppenni_node_fuerte.launch
```

Then run the command:

```
$ roslaunch rbx1_vision passthrough.launch
```

Next, bring up RViz with the provided Fuerte PCL config file:

```
$ rosrun rviz rviz -d `rospack find rbx1_vision`/pcl_fuerte.vcg
```

### Section 10.10.3 The VoxelGrid Filter

Remember to use the Fuerte versions of the `openni_node` launch file and the RViz config file:

```
$ roslaunch rbx1_vision oppenni_node_fuerte.launch
```

Then run the command:

```
$ roslaunch rbx1_vision voxel.launch
```

Next, bring up RViz with the provided Fuerte PCL config file:

```
$ rosrun rviz rviz -d `rospack find rbx1_vision`/pcl_fuerte.vcg
```

## Section 11.2 Object Tracker

Use the Fuerte version of the `openni_node` launch file:

```
$ roslaunch rbx1_vision oppenni_node_fuerte.launch
```

and the Fuerte version of the RViz config file:

```
$ rosrun rviz rviz -d `rospack find rbx1_nav`/sim_fuerte.vcg
```

### Section 11.2.4 Object Tracking on a Real Robot

**NOTE:** The stock `kinect.launch` file in the `turtlebot_bringup` package defaults to a relatively high video resolution of 640x480. Depending on the speed of your computer, this resolution can cause delays in the video stream and your robot's tracking will tend to oscillate. A better result is obtained by using a resolution of 320x240 pixels. You can either use `dynamic_reconfigure` to change the resolution or use the `kinect.launch` file in the `rbx1_bringup` package like this:

```
$ roslaunch rbx1_bringup kinect_fuerte.launch
```

### Section 11.3.1 Testing the Follower Application in Simulation

Use the Fuerte version of the `openni_node` launch file:

```
$ roslaunch rbx1_vision openni_node_fuerte.launch
```

and the Fuerte version of the RViz config file:

```
$ rosrun rviz rviz -d `rospack find rbx1_nav`/sim_fuerte.vcg
```

### Section 11.3.3 Running the Follower Application on a TurtleBot

**NOTE:** The stock `kinect.launch` file in the `turtlebot_bringup` package defaults to a relatively high video resolution of 640x480. Depending on the speed of your computer, this resolution can cause delays in the video stream and your robot's tracking will tend to oscillate. A better result is obtained by using a resolution of 320x240 pixels. You can either use `dynamic_reconfigure` to change the resolution or use the `kinect.launch` file in the `rbx1_bringup` package like this:

```
$ roslaunch rbx1_bringup kinect_fuerte.launch
```

### Section 12.1 A TurtleBot with a Pan-and-Tilt Head

Use the Fuerte version of the RViz config file:

```
$ rosrun rviz rviz -d `rospack find rbx1_description`/urdf_fuerte.vcg
```

### Section 12.4: Choosing a ROS Dynamixel Package

Instead of:

```
$ sudo apt-get install ros-electric-dynamixel-motor
```

use this:

```
$ sudo apt-get install ros-fuerte-dynamixel-motor
```

### Section 12.10.2: Monitoring the Robot in RViz

Use the Fuerte version of the RViz config file.

To view the robot in RViz and observe the motion of the servos as we start testing below, bring up RViz now as follows:

```
$ rosrun rviz rviz -d `rospack find rbx1_dynamixels`/rviz_fuerte.vcg
```

### Section 12.11.1: Tracking a Face

Use the Fuerte version of the `openni_node` or `uvc_cam` launch file:



```
$ roslaunch rbx1_vision openni_node_fuerte.launch
```

or if you are using a webcam:

```
$ roslaunch rbx1_vision uvc_cam_fuerte.launch device:=/dev/video0
```

(Change the video device if necessary.)

### Section 12.11.3: Tracking Colored Objects

Use the Fuerte version of the `openni_node` or `uvc_cam` launch file:

```
$ roslaunch rbx1_vision openni_node_fuerte.launch
```

or if you are using a webcam:

```
$ roslaunch rbx1_vision uvc_cam_fuerte.launch device:=/dev/video0
```

(Change the video device if necessary.)

### Section 12.11.4: Tracking Manually Selected Targets

Use the Fuerte version of the `openni_node` or `uvc_cam` launch file:

```
$ roslaunch rbx1_vision openni_node_fuerte.launch
```

or if you are using a webcam:

```
$ roslaunch rbx1_vision uvc_cam_fuerte.launch device:=/dev/video0
```

(Change the video device if necessary.)

## Section 12.12 A Complete Head Tracking ROS Application

Replace all instances of:

`rbx1_apps/head_tracker.launch`

with:

`rbx1_apps/head_tracker_fuerte.launch`

For example, the command:

```
$ roslaunch rbx1_apps head_tracker.launch face:=True
```

would become:

```
$ roslaunch rbx1_apps head_tracker_fuerte.launch face:=True
```

The only difference between the Electric and Fuerte versions of the launch file is that the Fuerte version includes the Fuerte versions of the `openni_node` and `uvc_cam` launch files.