A Gentle Introduction to ROS

(and related technologies)

"What are the requirements, how to install, how to setup, what are the parts of ROS, how to use on the most basic level"

- VM / Linux Installation
- Linux How-to
- ROS Installation & configuration
- Build a ROS project
- ROS concepts with example
- C++ OO concepts with example
- More ROS concepts

ROS Reference: "A Gentle Introduction to ROS" [Agitr]
Jason O'Kane http://www.cse.sc.edu/~jokane/agitr/

How Robotics Research Keeps...

Re-Inventing the Wheel

First, someone publishes...

...and they write code that barely works but lets them publish...

...a paper with a proof-of-concept robot.

This prompts another lab to try to build on this result...

...but they can't get any details on the software used to make it work...

But inevitably, time runs out...

...and all the code used by previous lab members is a mess.

So, a grandiose plan is formed to write a new software API...

...and countless sleepless nights are spent writing code from scratch.

Virtual Machine (VM) / Linux Install

Assume you run Windows/Mac. You need Ubuntu Linux; run in a VM

- Download Ubuntu Linux 14.04 (Trusty) desktop .iso
- http://www.ubuntu.com/download/desktop
- Download & install VirtualBox VM
- https://www.virtualbox.org/wiki/Downloads
- Start VirtualBox, create a new VM.
- Connect the virtual CD to your Ubuntu .iso download file. (Settings → Storage → (select CD drive) → CD icon dropdown → attach Ubuntu .iso
- Add a hard disk with 20GB (Settings → Storage → (select SATA controller & add hard disk) → (.vdi, dynamically allocated)
- Start VM → Install Ubuntu → answer Q's → restart VM when prompted. (OS will eject the CD)
- $ sudo apt-get update
- $ sudo apt-get upgrade

VM / Ubuntu Orientation

- VM: Normally close (X) the VM desktop & select “save to disk“ when shutting down the VM. Quick restart: select VM & Start.
- VM: Properties → Network: Start off with NAT. When you want to control a real robot, choose “Bridged“ network to give your VM a real address
- Ubuntu: Gear on right provides Ubuntu reboot
- After installing Ubuntu, Devices → Insert Guest additons CD, & run guest install, or if it doesn't run:
  - Start (Dash) → Terminal. $ /media/...VBOX.../autorun.sh
- Reboot to enable desktop resizing, copy/paste between host & guest
- Right click on launcher to unlock unneeded icons, lock terminal to launcher – you'll need it
- Dash → Appearance → Behavior. Show menu in title bar
- Dash → lock Turn lock off & passwd off
- Terminal: Ctrl-Shift-T: new tab. Ctrl-Shift-N: New window

Linux How-To

- References:
  - http://www.ee.surrey.ac.uk/Teaching/Unix/ Tutorials 1 & 2 for basic Linux commands (cd/pwd/ls)
  - http://linuxcommand.org/learning_the_shell.php
  - Google for Linux beginner Tutorial
- Note Firefox, Explorer in launcher
- Choose a code editor. Suggestion: sudo apt-get install kate
- In a terminal: Ctrl-Shift-C: copy; Ctrl-Shift-V: paste

Frequently Used Linux Commands

Command	Meaning
ls	list files and directories
ls -a	list all files and directories
mkdir	make a directory
cd *directory*	change to named directory
cd	change to home-directory
cd ~	change to home-directory
cd ..	change to parent directory
pwd	display the path of the current directory

Command	Meaning
cp *file1 file2*	copy file1 and call it file2
mv *file1 file2*	move or rename file1 to file2
rm *file*	remove a file
rmdir *directory*	remove a directory
cat *file*	display a file
less *file*	display a file a page at a time
head *file*	display the first few lines of a file
tail *file*	display the last few lines of a file
grep *'keyword' file*	search a file for keywords
wc *file*	count number of lines/words/characters in file

Start → Terminal
$ sudo apt-get update
$ sudo apt-get upgrade
Every few days!

$ sudo apt-cache search <topic>
$ sudo apt-get install <package>
$ ps -elf | grep ros
$ env | grep ROS

ROS Install

- [Agitr] sections 2.1 – 2.2
- http://wiki.ros.org/indigo/Installation/Ubuntu

$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu trusty main" > /etc/apt/sources.list.d/ros-latest.list'
$ wget https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -O - | sudo apt-key add -
$ sudo apt-get update
ONLY IF YOU'RE INSTALLING 14.04.2:
sudo apt-get install xserver-xorg-dev-lts-utopic mesa-common-dev-lts-utopic libxatracker-dev-lts-utopic libopenvg1-mesa-dev-lts-utopic libgles2-mesa-dev-lts-utopic libgles1-mesa-dev-lts-utopic libgl1-mesa-dev-lts-utopic libgbm-dev-lts-utopic libegl1-mesa-dev-lts-utopic
$ sudo apt-get install ros-indigo-desktop-full
$ sudo rosdep init
$ rosdep update
$ echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc
$ source ~/.bashrc
$ sudo apt-get install python-rosinstall

ROS Intro Tutorial Resources

- Agitr sections 2.3 – 2.8
- http://wiki.ros.org/ROS/Tutorials
- ROS by example Vols 1 & 2 P. Goebel
- Search the ROS wiki – there are a LOT of tutorials & other explanatory matter.

Make a place to build ROS projects

- Create and build an empty workspace

```
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/src
$ catkin_init_workspace
$ cd ~/catkin_ws/
$ catkin_make
```

- You should have no errors up to this point

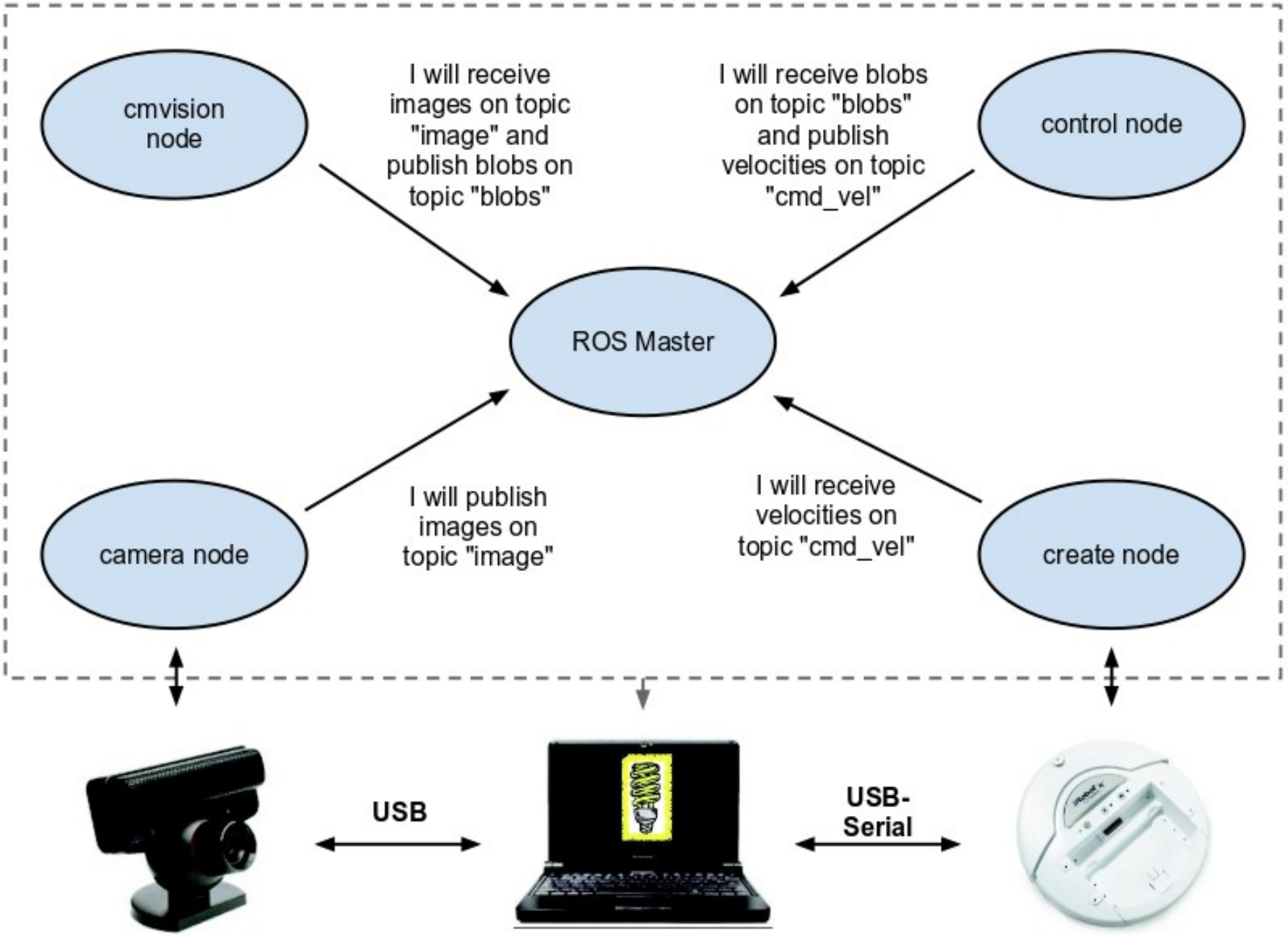
Download the agitr source code

- Download http://www.cse.sc.edu/~jokane/agitr/agitr-examples.zip
- Unzip it into ~/catkin_ws/src

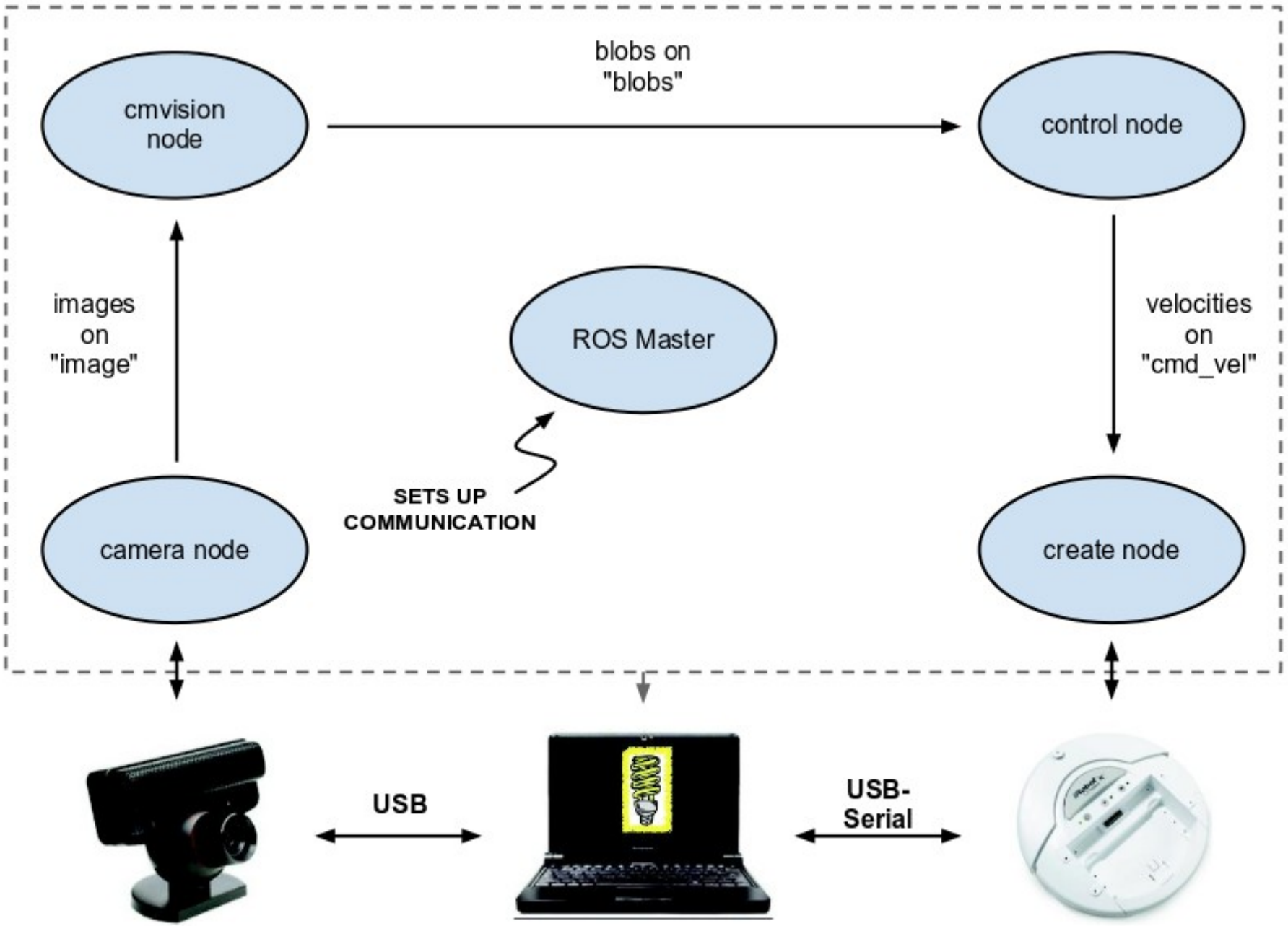
```
$ cd ~/catkin_ws
$ unzip ~/Downloads/agitr-examples.zip
$ catkin_make
```

- Now you can run the examples in the book

Review - How ROS works



Review - How ROS works



ROS Concepts & Demos

Topics, Publish/Subscribe

$ roscore
$ rosrun turtlesim turtlesim_node
$ rosrun turtlesim turtle_teleop_key
Move the turtlebot around. Now list topics & echo
$ rostopic list
$ rostopic echo turtle1/cmd_vel
Manually generate the cmd_vel msg
$ rostopic pub -l /turtle1/cmd_vel geometry_msgs/Twist -- '[2.0, 0, 0]' '[0, 0, 0.5]'

- rqt_graph
-

A little C++ / OO

- Classname myClass; is like int foo;
  - Syntax: Type name-of-instance-of-type;
  - Apple myApple(green); int foo=0;
  - Class names are capitalized, instance names are not
- Key distinction:
  - a class is a type. You can't say “I have Apple“ - doesn't make sense. You can say “ thisThing is an Apple“. Is-a relationship
  - An object is an instance of a type, distinct from all others
  - The apple in my bag is distinct from all other apples in the world (not necessarily different, but distinct (countable)
- Class (Type) represents a concept, of which there can be instances

A little C++ / OO

- Classes have properties and methods
  - Class Car has properties (attributes) like color, horsepower
  - Class car has methods like GasPedal(int percent)
    - Methods are a fancy name for functions that operate on the class

```
Car paulsCar("white");
paulsCar.GasPedal(100);
paulsCar.color = "fuscia";
```

- paulsCar is a different instance than jasonsCar

```
ros::Rate rate(2);
ros::Rate myRate = rate;     // resembles int i=0;
pub.publish(msg);
```

A little C++ / OO

- Namespaces: an ugly operator for a simple concept: scoping names

English – ROS Namespace

```
english::Fast
ros::Publisher
english::After
```

German - PaulsNamespace

```
german::Fast (means almost)
pauls_robot_ns::Publisher (different class)
german::After (means rectum)
```

```
ros::NodeHandle nh;
ros::Publisher pub;
```

A little C++ / OO

- Templates: a way of telling a class or method what class (Type-of-object) it's supposed to handle.
- Different than an argument (though you'd probably use an argument and a void pointer in C)

```
Value = objectName.functionName<Type>(args);
double d;
d = add<double>(1.0, 2.0);

NodeHandle nh;
pub1 = nh.advertise<msgType1>("topic1“, 1);
pub2 = nh.advertise<msgType2>("topic2", 1);


ros::Publisher pub=nh.advertise<geometry_msgs::Twist>("turtle1/cmd_vel",1000);
```

Publisher program

- [Agitr pg 46)

```
int main(int argc, char **argv){

    ros::init(argc,argv,"publish_velocity");
    ros::NodeHandle nh;


    ros::Publisher pub=nh.advertise<geometry_msgs::Twist>("
    turtle1/cmd_vel",1000);

    ros::Rate rate(2);
```

Publisher program

```
while(ros::ok()){
    geometry_msgs::Twist msg;
    msg.linear.x=2.0;
    msg.angular.z=0.5;

    pub.publish(msg);
    rate.sleep();
    }
}
```

Diag_demo

A demo was built to illustrate the power and use of the ROS diagnostic architecture. It simulates a GPS, camera & battery monitor, all emitting diagnostic status. Each HW simulator is a class with a method that's called at 100Hz and given a sinusoidally varying value with a period of about 10 seconds, which it uses to control its faulting behavior. It also demonstrates logging to file & console, and dynamic reconfiguration.

- https://github.com/PaulBouchier/diag_demo

Diag_demo

- Overall status, varying in real-time. rqt_robot_monitor
- Pop-out details for each monitored item
- Table of diagnostic data: rqt_runtime_monitor
- Text & ROS_INFO logs collected by the ros master. Display with rqt_console
- Dynamically altering log level - rqt_logger_level
- Echo /diagnostic & /diagnostic_agg
- Graph of ROS nodes: rqt_graph

So What Is ROS?

- Middleware that enables building a wide variety of robots out of fine-grained composable components
  - Middleware means IPC mechanisms, SW launcher, etc
- A community and their SW packages, offered up for anyone to use
- A development environment on Linux for building/running robot SW

Is ROS Right For Me?

Decent at 'C'? Comfortable working with big software systems? Open to learning enough Linux/Python? Want to build more capable robots than is possible from scratch? Open to using modules other ppl have developed (and learning enough to make them work for you?

- If Yes to all the above, ROS may be a good fit

ROS Cheat Sheet

Command Groups	Commands
Rosbash	*roscd <MyPackage>* rosls [location/subdir]
Packages	rospack find [packageName] catkin_create_pkg [packageName] [depend1] [depend2] rospack depends1 [packagename] - prints only 1st-order dependencies of packageName rospack depends [packagename] - prints all dependencies of packageName rosdep install packageName - installs dependencies of packageName rosdep check packageName - check dependencies catkin_make packageName - builds a package rosinstall - may install stuff - check tutorial catkin_init_workspace - initialize ros workspace - may install stuff
Nodes	*roswtf* - checks configuration consistency *rosnode list* - lists nodes rosnode info /rosout - prints publications etc of the target node rosnode ping /turtlesim - pings the node rqt_graph - draws graph of current connections between nodes nc -l 1234 listens for socket connection on port 1234, nc netbook 1234 talks on port 1234. Type to the talker & it should print on the listener.

ROS Cheat Sheet

Running	rosrun [package_name] [node_name] roslaunch [package_name] <launchfile> rosrun dynamic_reconfigure reconfigure_gui - reconfigures kinect
Topics	rostopic list[-v] [/topic/subtopic] - print info about active topic. -v is verbose rostopic pub <topic> std_msgs/String "br" - publish data to <topic> with data provided. e.g rostopic pub -l /turtle1/cmd_vel geometry_msgs/Twist -- '[2.0, 0, 0]' '[0, 0, 0.5]' rostopic echo - display data published to a topic rostopic type /topic - prints the message type of /topic rosmsg show message - prints the definition of a message. message can be <package> <msg> or just <msg> rossrv
Logging	*rosservice call /nodeName/set_logger_level <packageName> DEBUG* *rosbag [-O foo.bag]-* record topic1 topic2 Record published topics to a bag *rosbag record -a-* Record all topics *rosbag info foo.bag -* Print info about the bag, inc. start/stop times *rosbag play foo.bag* - Publish recorded topics
Data Visualization	*rosrun rviz rviz* - Set Fixed Frame to openni_rgb_optical_frame. Add a Point Cloud 2 display. Set topic to /camera/rgb/points. Note: first time you run it doesn't seem to allow adding anything. *rosrun image_view image_view image:=/camera/rgb/image_color* - displays camera image. *rxbag <bagfile>* - Display the contents of a bag. Also supports a record verb to record a bagfile. Has plugins for Image and Plot to plot numeric data. *rqt_plot /turtle1/pose/x,/turtle1/pose/y /turtle1/pose/theta* - Plot live data from topics
Parameters	rosparam list - lists the current parameters in the parameter server rosparam [get | set | load | dump] Call rosservice clear to activate new parameters rosparam get / shows values of all parameters rosparam dump params.yaml

ROS Cheat Sheet

Creating a catkin workspace:

$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/src
$ catkin_init_workspace
$ cd ~/catkin_ws/
$ catkin_make
# makes the empty workspace, creates devel & build folders
# catkin_make must be called in the top level of the workspace

Creating a catkin package

$ cd ~/catkin_ws/src
$ catkin_create_pkg beginner_tutorials std_msgs rospy roscpp
# i.e. command is catkin_create_pkg <pkgName> <dep1> <dep2>

Initialize a src area of workspace & download a pkg into it:

$ wstool init
$ wstool set ros_tutorials --git git://github.com/ros/ros_tutorials.git
$ wstool update
Alternatively, just pull it from git with:
$ git clone git://github.com/ros/ros_tutorials.git

Resources

- Documentation. The authority, with tutorials, API docs etc. http://wiki.ros.org/
- Answers. Ask technical questions, get answers
  - http://answers.ros.org/questions/
- Books.
  - A Gentle Introduction to ROS. Jason O'Kane
  - Ros By Example. Patrick Goebel