

ROS Introduction - Practical

Patrick Nagel

University of Applied Sciences Bonn-Rhein-Sieg

March 27, 2018



**Hochschule
Bonn-Rhein-Sieg**
University of Applied Sciences

Agenda

- 1 Setting up a new ROS Project
- 2 Navigation Tips
- 3 Hello World
- 4 Analysis Tips
- 5 Configuring an existing ROS Project



**Hochschule
Bonn-Rhein-Sieg**
University of Applied Sciences

Setting up a new ROS Project

- Create a directory of the following structure:
mkdir -p /.../<workspace_name>/src
- In root of the workspace run: **catkin_make**
- Initialize your workspace: **catkin init**
- Set up your working environment:
source /.../<workspace_name>/devel/setup.bash
- If workspace default workspace, then add the line to your
~/.bashrc



Navigation Tips

Prerequisite for these tips is that the workspace of interest overlayed the default path in `$ROS_PACKAGE_PATH`:

e.g. `/.../<workspace_name>/src/opt/ros/kinect/share`

- Find the path to a package:
`rospack find <package>`
- Navigate to your workspace or a package within:
`roscd [<package>]`
- Navigate to the log files of your project:
`roscd log`
- Use the *TAB* key for autocompletion.



Hello World

- Create a package within `<workspace_name>/src`:
`catkin_create_pkg <package_name> [depend1] [...]`
- Create a python file within the package and add code.
- Make the python file executable: **`chmod +x <python_file>`**
- Build the project together with the new package. In root:
`catkin_make`
- Start *roscore* and execute:
`roslaunch <package_name> <python_file>`
- Echo published message:
`rostopic echo /<publisher>`



Hello World - Code

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-

import rospy
from std_msgs.msg import String

def talker():
    pub = rospy.Publisher('chatter', String, queue_size=10)
    rospy.init_node('talker', anonymous=True)
    rate = rospy.Rate(10)
    while not rospy.is_shutdown():
        hello_str = "hello world %s" % rospy.get_time()
        rospy.loginfo(hello_str)
        pub.publish(hello_str)
        rate.sleep()

if __name__ == '__main__':
    try:
        talker()
    except rospy.ROSInterruptException():
        pass
```



Analysis Tips

Prerequisite for these tips is that *roscore* and the nodes you are interested in, are running.

- Display configuration details of your workspace:

catkin config

- Display debug information about ROS topics:

rostopic

- **list**: Displays a list of current topics.
- **echo <topic>**: Displays messages published to a topic.
- **info <topic>**: Print information about a topic.
- **pub <topic> <msg_type> <data>**: Publish data to a topic.

- Display debug information about ROS nodes:

roscnode

- **list**: Display a list of current nodes.
- **info <node>**: Print information about the chosen topic.

- List and query ROS services:

rosservice [list], [info <service>], ...

- Get and set ROS parameters:

rosparam [set], [delete] <param_name> <value>

- List package dependencies:

rospack depends1 <package>

Configuring an existing ROS Project - e.g. b-it-bots

- Navigate to the directory in which you would like to set up the project and create the ROS workspace structure:
mkdir -p <workspace_name>/src
- Copy the link of the git repo(s) and run inside the **src** folder:
git clone <link>
- Compile the workspace, by navigating to root and executing:
catkin_make
- If errors appear install the missing packages and rerun
catkin_make

