AI & Robotics

Creating a Catkin Package

Goals

The **junior-colleague**
- can describe a catkin workspace in own words
- can explain the package structure of a catkin Workspace
- can create a package with dependencies inside a catkin workspace
- can inspect the dependencies of a ROS Package

Catkin Workspace

- Directory

- Modify, build, and install catkin packages

- Catkin package = standalone (sub)project

- The Virtual Machine has one workspace

  `/home/user/Projects/catkin_ws` (For your packages.)

Catkin Workspace

- First time install

```
$ cd ~/Projects/catkin_ws
$ catkin_make
$ echo "source ~/Projects/catkin_ws/devel/setup.bash" >> ~/.bashrc
$ source ~/.bashrc
```

Contents of a Catkin Package

- Simplest possible package (in theory!):

  ```
  $ ls own_package/
  CMakeLists.txt
  package.xml
  ```

- Catkin compliant `package.xml`

- `CMakeLists.txt` which uses catkin

- Max. 1 package in each folder
  (No nested packages or multiple packages)

Package Structure i/a Catkin Workspace

```
workspace_folder/ -- WORKSPACE
   src/ -- SOURCE SPACE
      CMakeLists.txt -- 'Toplevel' CMake file, by catkin
      package_1/
         CMakeLists.txt – CMake file for package_1
         package.xml -- Package manifest for package_1

      ...

      package_n/
         CMakeLists.txt -- Cmake file for package_n
         package.xml -- Package manifest for package_n
```

Creating a Catkin Package

```
$ cd ~/Projects/catkin_ws/src
$ catkin_create_pkg first_package std_msgs rospy
Created file first_package/package.xml
Created file first_package/CMakeLists.txt
Created folder first_package/src
Successfully created files in /home/user/Projects/catkin_ws/src/first_package.
Please adjust the values in package.xml.
$ ls first_package/
CMakeLists.txt  package.xml  src
$
```

- Syntax:
  catkin_create_pkg <package_name> [depend1] [depend2] [depend3] …

- More info:
  http://wiki.ros.org/catkin/commands/catkin_create_pkg

First-Order Dependencies

```
$ rospack depends1 first_package
[rospack] Error: the rosdep view is empty: call 'sudo rosdep init' and 'rosdep update'
$ sudo rosdep init
[sudo] password for user:
ERROR: default sources list file already exists: /etc/ros/rosdep/sources.list.d/20-default.list
Please delete if you wish to re-initialize
$ sudo rm /etc/ros/rosdep/sources.list.d/20-default.list
$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

    rosdep update
$ rosdep update
 . . .
$ rospack depends1 first_package
rospy
std_msgs
$
```

First-Order Dependencies

```
$ rospack depends1 first_package
rospy
std_msgs
$ roscd first_package
$ cat package.xml
<package>
    ...
    <buildtool_depend>catkin</buildtool_depend>
    <build_depend>rospy</build_depend>
    <build_depend>std_msgs</build_depend>
     ...
    <run_depend>rospy</build_depend>
    <run_depend>std_msgs</build_depend>
    ...
</package>
```

First-Order Dependencies

```
$ rospack depends1 rospy
genpy
roscpp
rosgraph
rosgraph_msgs
roslib
std_msgs
```

- Dependencies have also dependencies
- rospack can recursively find all dependencies
  (Next slide.)

Indirect Dependencies

```
$ rospack depends first_package
genmsg
genpy
cpp_common
rostime
roscpp_traits
roscpp_serialization
message_runtime
catkin
gencpp
genlisp
message_generation
rosbuild
rosconsole
std_msgs
rosgraph_msgs
xmlrpcpp
roscpp
rosgraph
rospack
roslib
rospy
```

Customizing a Package: `package.xml`

```xml
<package>
    <name>first_package</name>
    <version>0.1.0</version>
    <description>The first_package package</description>
    ...
    <maintainer email="user@todo.todo">user</maintainer>
    ...
    <license>TODO</license>
    ...
    <buildtool_depend>catkin</buildtool_depend>
    ...
    <build_depend>rospy</build_depend>
    <build_depend>std_msgs</build_depend>
    ...
</package>
```