ROS Build System

ARRA / AR2A

Advancements for **R**obotics in **R**escue **A**pplications

March 13, 2016

1. rosbuild[1]

2. catkin[1]

ARRA/AR2A

What do we want?

- ARRA / AR2A aims to improve the current state of technology of robotics in rescue applications.

Who are we?

- A volunteer non-profit organisation of robotic enthusiasts.

How can you help?

- Check us out at `https://github.com/ar2a`

License information

Introduction 1

- Software in ROS is organized in packages
- It provides a way to easily reuse ROS-Software

A ROS package might contain:

- ROS nodes
- a ROS-independent library
- a dataset
- configuration files
- a third-party piece of software
- or other useful modules that can be grouped together logically

Introduction 2

Why does ROS have a custom build system?

For development of single software projects, existing tools like Autotools, CMake, and the build systems included with IDEs tend to be sufficient.

ROS is a very large collection of loosely federated packages. That means lots of independent packages which depend on each other, utilize various programming languages, tools, and code organization conventions.

Section 1

rosbuild[1]

Utilities

Tools for installing, building and locating of packages

- Build: rosmake
- Install: rosinstall
- Searching for packages: roslocate
- Install thirdparty libraries: rosdep
- Packages: rospack, roscd
- Stacks: rosstack, roscd

Build: rosmake

- rosmake is a tool to assist with building ROS packages. It facilitates building packages that have dependencies.
- ROS comprises a large number of packages. With the exception of some core packages that everything else depends on, many of the packages are largely independent. ROS provides build system to allow to only build what is actually necessary to run the packages.
- A package may depend on any number of other packages, requiring that those packages be built first. These dependencies are specified in the package's manifest.xml file.
- rosmake do the ROS-wide build of everything need for a given package.

Example: build move_base

```
$ rosmake move_base
```

rosmake will determine move_base's dependencies (and the dependencies' dependencies), and will ensure that all dependencies are built prior to building move_base.

- http://wiki.ros.org/rosmake

R.I.P. rosbuild

Section 2

catkin[1]

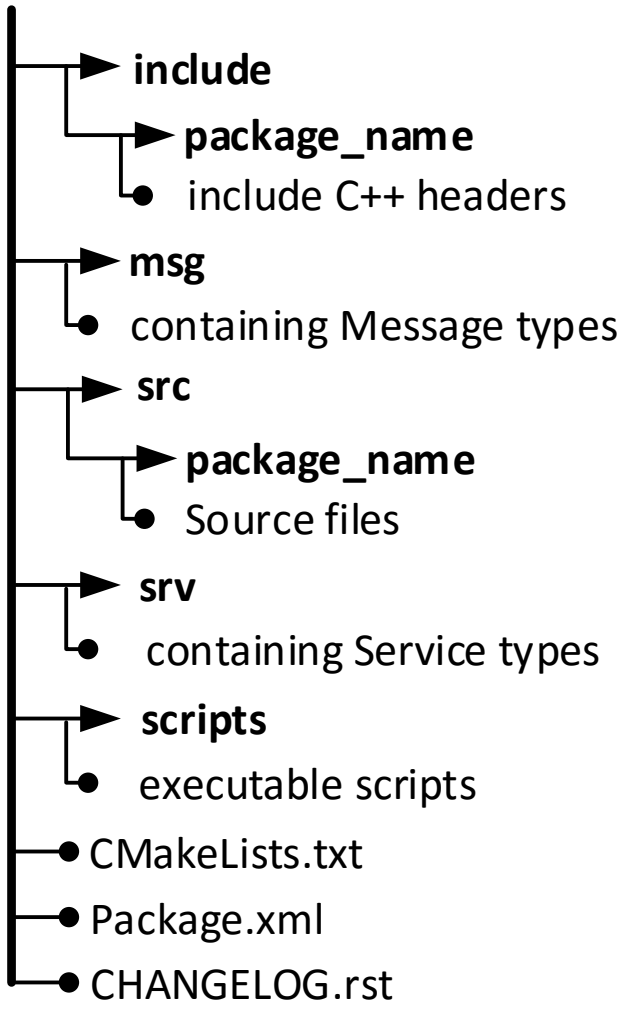
What is Catkin?

Definition

Catkin was designed to be more conventional than rosbuild, allowing for better distribution of packages, better cross-compiling support, and better portability.

- Catkin is the official build system of ROS (which replaces rosbuild)
- It combines CMake macros and Python scripts to provide some functionality on top of CMake's normal workflow
- Support for automatic 'find package' infrastructure and building multiple, dependent projects at the same time

General structure of a ROS Package



Setup the built environment

Creating a workspace

```
$ source /opt/ros/indigo/setup.bash
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/src
$ catkin_init_workspace
```

Build empty project

```
$ cd ~/catkin_ws/
$ catkin_make
```

Environment-Variables loaded?

```
$ source devel/setup.bash
$ echo $ROS_PACKAGE_PATH
/home/youruser/catkin_ws/src:/opt/ros/indigo/share:
/opt/ros/indigo/stacks
```

Create and Build a ROS Package

Create a new package called beginner_tutorials which depends on std_msgs, roscpp, and rospy

```
$ catkin_create_pkg beginner_tutorials std_msgs rospy roscpp
```

Build a package

```
$ cd ~/catkin_ws/
$ catkin_make
```

Eclipse and ROS 1

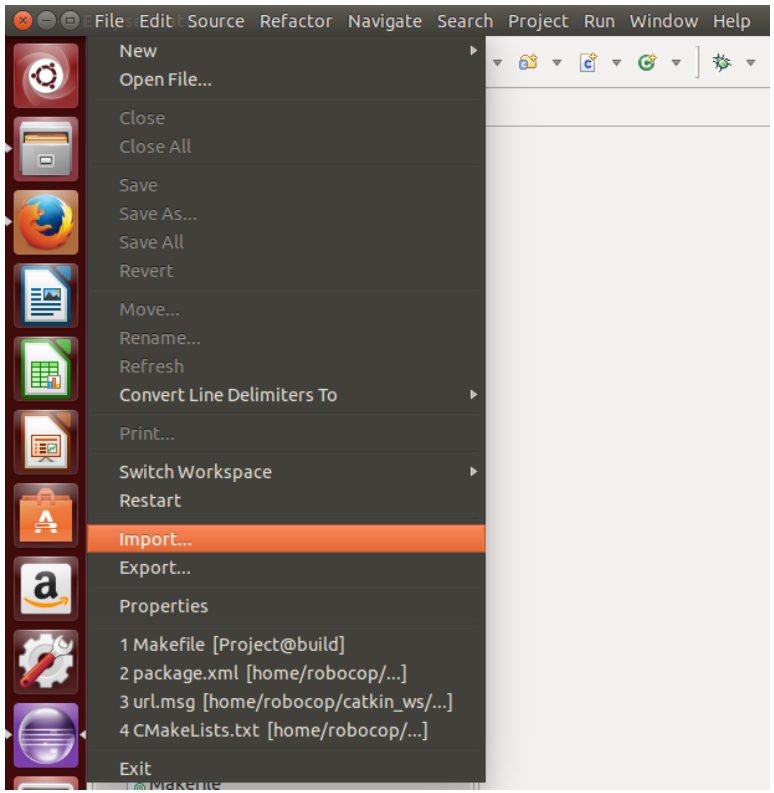
Creating the Eclipse project files

```
$ cd ~/catkin_ws
$ catkin_make --force-cmake -G"Eclipse␣CDT4␣-␣Unix␣Makefiles"
$ cd build
$ awk -f $(rospack find mk)/eclipse.awk build/.project >
build/.project_with_env && mv build/.project_with_env
build/.project
```

Importing the project into Eclipse

- Start Eclipse, select File –>Import. Select Existing projects into workspace, hit next, then browse for your package's directory (select root directory). Do NOT select Copy projects into workspace. Then finish.

Importing the project into Eclipse



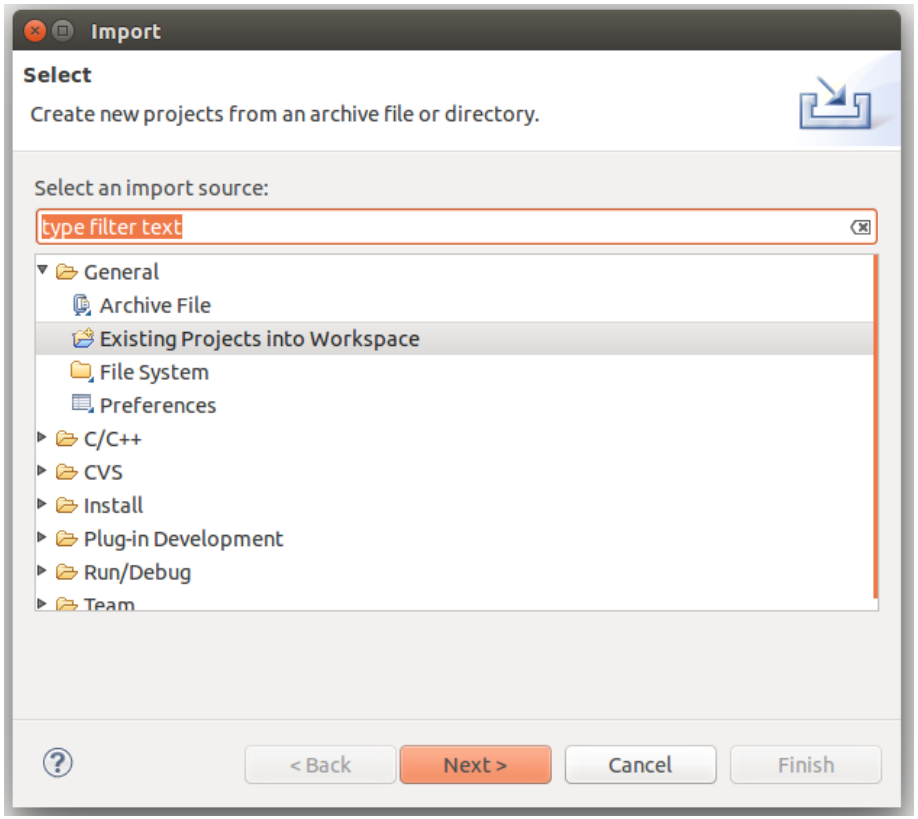


Figure: Start Eclipse, select File –>Import. Select Existing projects into workspace, hit next.

Eclipse and ROS 3

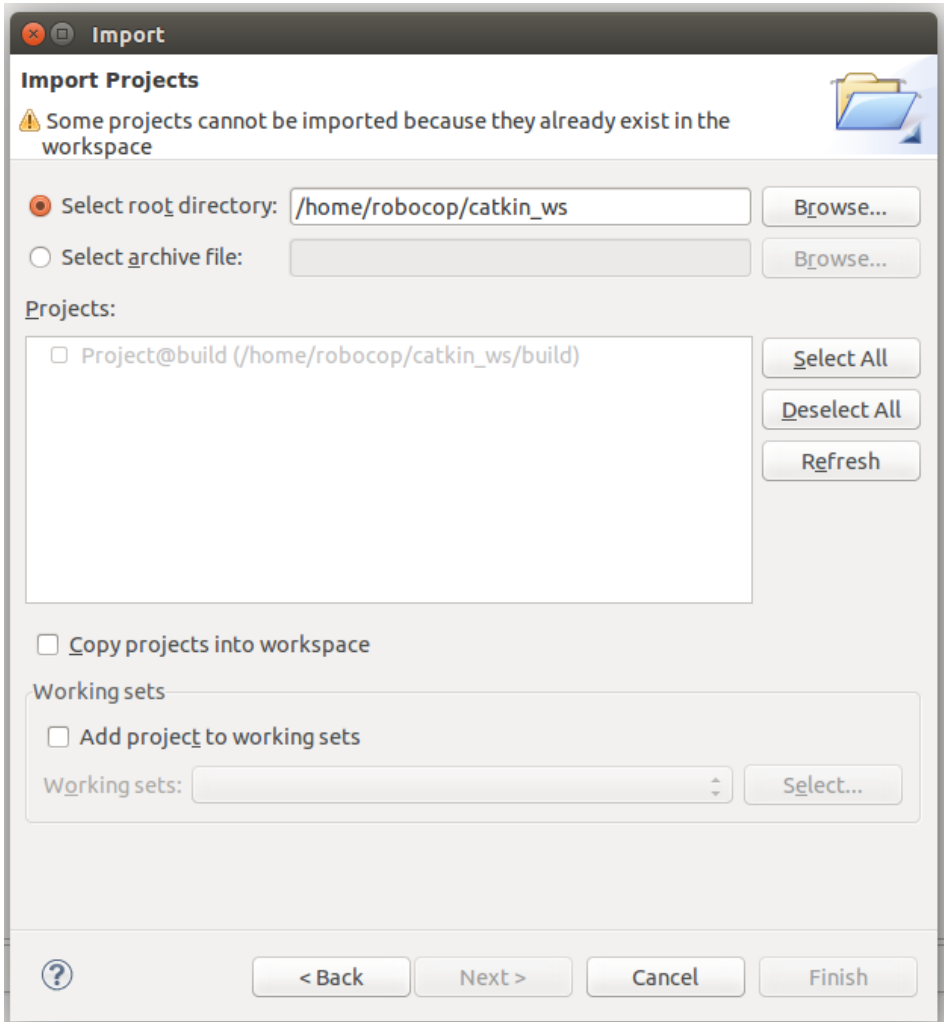


Figure: Browse for your package's directory (select root directory). Do NOT select Copy projects into workspace. Then finish.

References I


"Ros wiki," 11 2015.
http://www.ros.org/.

The End