

Overview

These are initial notes drawn from the birds of a feather meeting at RosCon 2013.

Core Issues

- Easily working with a multi-repo rosjava and android workspace (sequencing builds from start to finish like catkin does)
 - For rosjava repos, gradle superprojects don't cross-references across repo boundaries
 - For android repos, hardcoded relative referencing to libraries across repos doesn't work
- How to make best use of maven for rosjava and android packages.
- Packaging rosjava repositories (not android repos - worry about these later)

Discussions

General Topics

****Branch development****

- Move all new development to hydro branches, leave groovy frozen while we experiment.
- Damen prefers groovy, hydro labels, but it conflicts with defact standard *-devel notation used by rest of ros.

****Java/Android Namespaces****

- Damon's advice to take care when naming these (e.g. org.ros.android), follow the [documentation guidelines](#).

If you don't have an official url you can use to guarantee uniqueness - use your repo's github url, e.g. com.github.stonier.rosjava_wunderbar.

****Pull Requests to Damon's Repos****

- First sign the CLA
 - Individual: <http://code.google.com/legal/individual-cla-v1.0.html>
 - Company: <http://code.google.com/legal/corporate-cla-v1.0.html>

****Style Guide****

Follow damon's style guide!

Java Topics

****Packaging****

Need alot more detail to understand how this could actually work, but damon suggested to work it into bloom somehow (as opposed to packaging everything in an internet maven repository).

- Uses a system level maven cache per rosdistro (does it really need to be rosdistro based?).
- Gradle target installs the package
- Have a closer look at this. Depending on options, feed that into bloom.
- All the ros package jars should go into classpath and live in our rosdistro maven.

****Rosdep Jars****

Rosdep kinds of jar dependencies are currently in robotbrain's maven site (a rosjava user with swi).

We need our own way of doing this - either by replacing robotbrain with access to our own maven site or distribute these jars in debs.

****Java Build System****

We were discussing two ways of automating java package builds across repos.

- drop in a build.gradle at the root of your multi-repo workspace
- use catkin, some cmake and packge.xml build depends to resolve and build java packages sequentially

The first is simpler, and doesn't require the black magic of involving catkin, but...it means you can't build alongside with other catkinized repos you might add your source workspace (e.g. msg packages!).

****Rosjava Jmdns****

Damen mentioned he would like to see this in rosjava_core. Prepare the package and some

tutorial programs.

Android Topics

****RosActivity****

Everyone use RosActivity so we can converge fixes in RosActivity. We should make sure this is in documentation.

****Rosdep style Jar Dependencies****

Sometimes these are custom built jars that aren't available from usual maven repositories. A good example is snakeyaml that kazuto used - the maven central version of snake yaml is not compatible with android, but they do supply a jar file that is stripped down and compatible with android.

Have a gradle script take the web download and install to the local maven cache (maybe a package to do this?). Is it not possible to have in a maven repository and simply add that repository to the build.gradle file?

**Update:* this is apparently a non-issue, it should be possible to use our own maven repo to pull jars for android packages from.

****Why is gradle so slow for android****

Dexing everything, for each package. It is also building the apk file, whereas eclipse incremental builds only build the classes on the fly.

****Preferred development environment****

Eclipse - far more useful and faster. Current workflow is to build everything with gradle, then import android code and develop inside eclipse. Note, if you touch a msg file, then you have to clean and rebuild everything from rosjava_messages and up otherwise jar files mismatch. That's a really long process.

****OSRF Android App Patches****

Minor patches into android_core, app manager stuff elsewhere (probably rocon_android for turtlebot

hydro development for now (see below comment).

****Turtlebot/PR2 Apps****

We will really branch this for turtlebot hydro as we'll upgrade the turtlebot 1-1 and multimaster connections to android tablets. Reaction is just to freeze groovy development for both as is and go ahead with the turtlebot on its own.

****Android Build Environment****

I was working on a way to fix the real problem with android builds - that of automatically running the android update project call as well as resolving relative library dependencies via catkin. This was working well - it used package.xml and build_depends and would let you build a chain of android packages across repos via the usual catkin_make call (or similar).

However - google has just released a new gradle plugin which is intended to replace the adt.

* <http://tools.android.com/tech-docs/new-build-system/user-guide>

Don't know much about it yet. Some key points, it will replace adt and will integrate with the ide development process (currently adt and eclipse are incompatible in various ways).

We should really try this and see what advantages it offers. High priority!

.. _documentation guidelines:

http://rojava.github.io/rojava_core/best_practices.html#java-package-names