New Project Checklist

Introduction

Getting started in OpenDaylight can be a bit intimidating, but most of the infrastructure vanishes into the background pretty quickly as you get things set up and used to the workflow.

New Project Checklist

Here are the tasks you're going to want to do

1. Make sure you have a git repository (visible in gerrit)
    a. Go to https://git.opendaylight.org/
    b. Click "Projects" on the left under the OpenDaylight logo
    c. Click "List" under projects
    d. Verify your project is listed with the right repo name
2. Make sure your Jenkins jobs are created
    a. Go to https://jenkins.opendaylight.org/releng/
    b. Look for a tab at the top for your project and click on it
    c. Make sure you have at least <repo-name>-verify-master, <repo-name>-merge-master, and <repo-name>-daily-master jobs
    d. If the jobs do not exist follow the instructions from the releng/builder Jenkins documentation to configure your jobs. RelEng/Builder Jenkins page
3. Verify that your jenkins jobs work
    - Test verify by pushing a patch and seeing that the verify job runs and successfully posts a "verified +1" to the gerrit.
    - Then merge the patch and make sure that the merge job runs successfully and pushes artifacts in Nexus.
        - After the job runs, go to http://nexus.opendaylight.org/
        - Search for your project's name and see if the artifacts you expect are there.
    - Optionally, test your integration jobs by manually triggering them.
4. Make sure you have a JIRA entry for your project
    a. Go to JIRA
    b. Make sure your project appears in the list
5. Subscribe to your mailing list and other important mailing lists
    a. Go to https://lists.opendaylight.org/
    b. Confirm that you are subscribed to either kernel-dev or app-dev mailing lists
6. Subscribe to these other suggested lists:
    a. tsc
        i. At least one representative from your project should either (i) attend the TSC calls, (ii) subscribe to the TSC mailing list, or (iii) read the TSC meeting minutes.
        ii. Ideally, a project representative should do all three.
    b. dev
    c. discuss
    d. Your gerrit so you can tell when people push patches to your project (see LF Gerrit Guide for how to subscribe to events from your project)
    e. The Jenkins mailing list, filtered for the topic equal to your project name. Test failures will be reported to this list.
7. Set up your pom file structure according to project layout best practice
    - A good way to do this to use the Project Startup Archetype, but replace the example module and hello world RPC with the beginnings of your own project
    - Your project should have a parent pom file that should be the root of your 'pom tree'
    - It's parent should be the odlparent pom file
    - All version numbers should be put into one of these two pom files, not hard-coded in other pom files
    - Default Version numbers from the "Startup Project Archetype" are not up to date. So, please update the version numbers in all the files generated.
        - Example to change the version numbers - https://git.opendaylight.org/gerrit/#/c/28662/
    - Certain pom files should use special parent poms, e.g., the karaf-parent in controller to build Karaf distributions. Other useful parent pom files to know about are:
        - binding-parent in yangtools for defining YANG models
        - features-parent in odlparent for defining features repositories
        - karaf-parent in controller for defining project-local Karaf distributions
8. Set up your JaCoCo code coverage reporting
    - Please follow these simple steps to report unit and integration test code coverage to Sonar
    - Currently, this only works for Java

Taking Advantage of Project Infrastructure

There is a detailed overview of the Development Infrastructure that will give you a good conceptual framework. Most of what is written here is about concrete getting things done stuff :)