

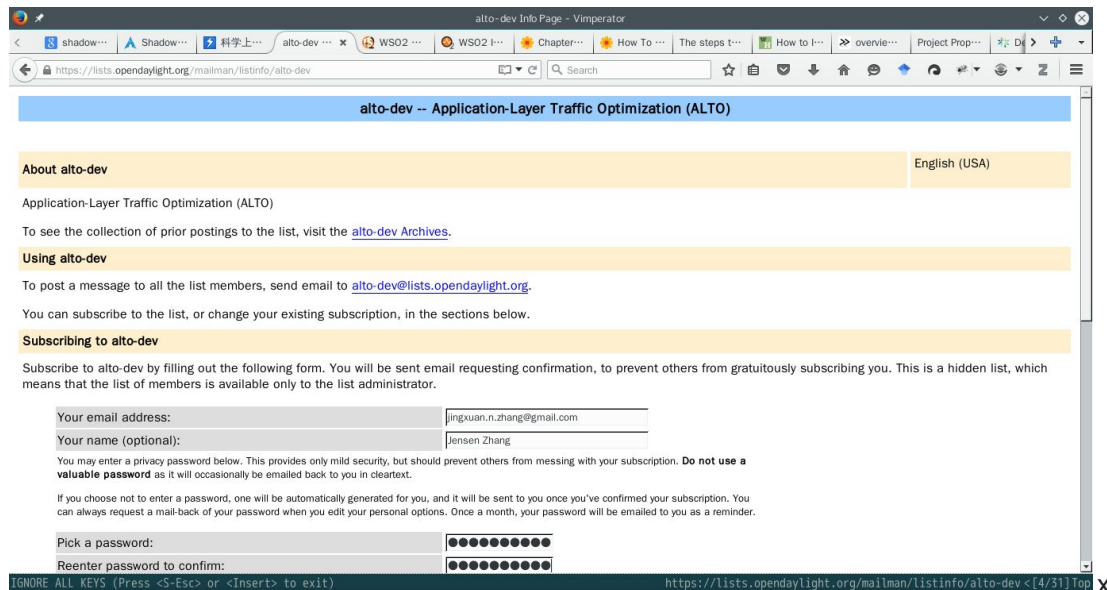
Part 1: Work Environment Setup

Create your account in Opendaylight Gerrit

- [ODL Gerrit Setup Guide](#)
- [ODL Gerrit Link](#)

Subscribe your email to [alto-dev](#) email list

- [Link of All ODL Public Mailing Lists](#)
- [Link for subscribing alto-dev mailing list](#)



The screenshot shows a web browser window displaying the 'alto-dev -- Application-Layer Traffic Optimization (ALTO)' mailing list page. The page has a blue header bar with the title. Below the header, there's a section titled 'About alto-dev' with a link to 'alto-dev Archives'. Another section titled 'Using alto-dev' provides instructions on how to post a message to the list. The main section is 'Subscribing to alto-dev', which contains a form for subscription. The form includes fields for 'Your email address' (filled with 'jingxuan.n.zhang@gmail.com'), 'Your name (optional):' (filled with 'Jensen Zhang'), and a 'Pick a password' field with a masked input. A note explains that a privacy password is recommended for security. At the bottom, there's a 'Reenter password to confirm:' field. The browser's address bar shows the URL 'https://lists.opendaylight.org/mailman/listinfo/alto-dev'.

- If you are interested in other ODL projects and hope to get latest information from that project, go ahead and subscribe to accordingly mailing list.

Clone ALTO Project from ODL Gerrit

- You'd better add your ssh-key to the gerrit of ODL. Although it's not necessary to clone a repo from gerrit, if you want to contribute and push your code via git, a ssh-key is needed. Check [ODL Wiki](#) for details.
- People in China may not be able to get access to ODL Gerrit repo because of GFW. You can use Github instead. However, you shall notice that they are just mirrors of projects in ODL gerrit, which means you shall not push code to Github directly.
 - [Opendaylight Github Link](#)
 - [ODL ALTO Github Link](#)

Compile and Install ALTO

Prerequisites

- [Development Environment Setup for ODL](#)
 - **Important:** Don't forget to edit your `~/.m2/settings.xml` file otherwise you will not be able to download any ODL Java dependencies

Steps

- Enter your ALTO project root directory
- Checkout to `stable/lithium` branch and run ***mvn compile*** and ***mvn install*** to compile and install ALTO

Notes:

- If you met any problems while compiling or installing, check **Common Problems List** section at the end of the doc first. If it does not help, please email godrickk@gmail.com, dongs2011@gmail.com or 2010_fno@tongji.edu.cn for help.

Part 2: ALTO Introduction

ALTO Protocol

The full name of ALTO is **Application-Layer Traffic Optimization**. Read [RFC 7285](#) to get more details.

Basic Components

[ALTO in ODL Implementation Introduction Slides](#)

Components	Description
ALTO Yang Model	A YANG Model of data instances and RPCs to support ALTO.
ALTO Provider	A simple implementation of endpoint cost map service.
ALTO Hosttracker	An implementation of automatically generating network maps and cost maps using collected network host information (by I2switch).
ALTO Services	Implementations of RFC 7285 format map services.
ALTO Northbound	RESTful API based on ALTO protocol.
ALTO Manger	Supports of map managements.
ALTO Features	
ALTO Commons	Common library.
ALTO Config	Configs to initiate ALTO services in Karaf.
ALTO Karaf	Create e karaf instance with all ALTO features installed

Build and run ALTO in Karaf

- Checkout to `stable/lithium` branch
- Run `mvn install` to build the project

- Enter into `alto-karaf/target/assembly/` directory
- Run command `./bin/karaf` to start the karaf
 - It will take some time to install features, initialize services and download dependencies. Be patient and check `./data/log/karaf.log` file to track the latest status.
- Play around. You can:
 - use ALTO Manager load/unload maps from/to ODL data store
 - use RESTCONF to read/insert data into ODL data store
 - query map service through ALTO Northbound API
 - Check [ALTO User Guide](#) for more details

Import ALTO to Eclipse

- Run `mvn eclipse:eclipse` to generate Eclipse IDE files (*.classpath, *.project ...)
- Import ALTO project to Eclipse as general Java projects
- Check [ALTO Developer Guide](#) for more details.

Part 3: Useful Docs and Links

Open Source Tools Used by Opendaylight Project

[IRC](#): Communicate with other ODL members

[Bugzilla](#): Submit and track bugs related to the OpenDaylight project.

[SonarQube](#): Help review and manage the code quality of the OpenDaylight project.

[Gerrit](#): Access Git repositories, review, and comment on the OpenDaylight code base.

[Jenkins](#): Integrate changes or obtain a fresh build of components.

[Nexus](#): Proxy remote repositories and share software artifacts.

[Ask](#): Q&A Forum for Developers and Users.

[Wiki](#): Opendaylight wiki for developers

Opendaylight and ALTO Release

[ODL Release Plan](#)

[ODL Release Status Tracker\(Google Doc\)](#)

[ALTO Project Proposal](#)

[ALTO Release Plan](#)

[ALTO Release Review](#)

[ALTO Release Note](#)

Part 4: Common Problem List

N/A yet