

# Android for Project Ara: Getting Started Guide

## Revision History

release 0.10 - 6/26/14 - initial public release

release 0.11 - 6/30/14 - added `install build-essential` to Common Installation

## Overview

This document describes how to get up and running on Android Platform development for Project Ara.

**Android builds take a long time.** You want a nice SSD and the fastest machine with the most RAM you can get your hands on.

## Setup Instructions

These instructions explain everything you need to do to go from a fresh Ubuntu 12.04, 13.10, or 14.04 (64-bit only) installation to an Android source tree with binaries you can flash to an SD card.

Additions to this document for other platforms are welcome. Any missing steps are bugs; please report them so we can update this document.

- **Step 1: install dependencies.**

- # Common installation (Ubuntu 12.04 and later)**

- ```
sudo apt-get install m4 git-core flex gperf automake texinfo bison
sudo apt-get install build-essential
```

- ```
sudo add-apt-repository ppa:linaro-maintainers/tools
sudo apt-get update
sudo apt-get install linaro-image-tools
```

- # Ubuntu 12.04 only**

- ```
sudo apt-get install ia32-libs
```

- # Ubuntu 13.10 and later**

- ```
sudo apt-get install libc6:i386 libstdc++6:i386 zlib1g:i386
```

- # Ubuntu 14.04 only**

- ```
sudo apt-get install libswitch-perl u-boot-tools
```

- **Step 2: set your name and email address for Git.**

```
git config --global user.name "Your Full Name Here"
git config --global user.email "your-email@example.com"
```

- **Step 3: download the repo tool, and ensure it's in your PATH.**

```
mkdir ~/bin
cd ~/bin
wget http://commondatastorage.googleapis.com/git-repo-downloads/repo
chmod +x ~/bin/repo
export PATH=~/bin:$PATH
```

- **Step 4: download the sources.** Get the sources with:

```
cd /directory/where/you/want/android
```

```
export MFST_REPO=https://ara-mdk.google.com/manifest
export MFST_BRANCH=master
export MFST_FILE=google-ara-dev.xml
export REPO_GROUPS=common,ara-dev
```

```
repo init -u $MFST_REPO -b $MFST_BRANCH -m $MFST_FILE -g $REPO_GROUPS
# The following "repo sync" might not work if you need to enter your
# key's passphrase. To avoid this with passphrase-protected keys,
# use ssh-agent and ssh-add (as described in this ssh-agent HOWTO).
repo sync
```

The "repo sync" call will download the latest sources. This will take a while.

- **Step 5: get Oracle JDK 6.**

<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html>

Make sure to set JAVA\_HOME to point to the installed JDK directory, and ensure that its bin/ directory is on your PATH.

You may be able to get away with later JDK versions (and potentially OpenJDK) by commenting out the relevant makefile version checks, but doing that is up to you.

- **Step 6: set up ccache.** This caches object files to speed up rebuilds. See [ccache instructions here](#).
- **Step 7: Start development!** Begin by [building Android](#), then [flash an image onto an SD card](#). You'll be able to boot this card once Project Ara development boards are released.

## Development Cheat Sheet

This section provides HOWTOs for basic development tasks. They assume you're running Ubuntu; you may have to change some of these instructions (e.g. for systems that don't use Upstart) on other operating systems.

### Set Up ccache

Android takes a long time to build. You should use ccache to save object files for later rebuilds. The virtual machine image already has ccache set up. Read [the Android ccache guide](#) to adjust the ccache configuration, or set it up on your own machine.

### Build An Android Image for an SD Card

To build an image, run these from the Android root directory:

#### **# Prepare the environment**

```
export CDPATH=
export TARGET_PRODUCT=pandaboard
export \
TARGET_TOOLS_PREFIX=prebuilts/gcc/linux-x86/arm/arm-linux-androideabi-4.7/bin/
arm-linux-androideabi-
. build/envsetup.sh
```

#### **# Build the image's partitions' contents as tarballs.**

```
make -j12 boottarball systemtarball userdatatarball
```

You only need to prepare the environment once per shell session. You can then call make as many times as needed.

Replace "12" in the -j argument to make to suit your system. On a machine with N cores (or N threads, for hyperthreaded CPUs), the usual recommendation for -j is a number between N and 1.5N.

The tarballs go in the build directory, which is "out", relative to the Android source root. These are the contents of the boot, system, and user data partitions which you will install on the SD card next.

### Install an Android Image on an SD Card

First, build the Android image's partition tarballs as [described above](#). (Having a valid TARGET\_PRODUCT in your environment is important for this section.)

Then you need to figure out the device node for the SD card you wish to install on. Many USB-to-SD dongles present as /dev/sdX, for some letter X. Other SD interfaces present /dev/mmcblkN, for some number N.

Once you know your device node, run the following from the root of your Android tree (the call to `linaro-android-media-create` should all be on one line):

```
# BE 100% SURE ABOUT THE VALUE OF "mmc".
# THE WRONG CHOICE CAN OVERWRITE YOUR HARD DRIVE.
#
# Use the letter-only /dev entry, e.g. "/dev/sdf", NOT "/dev/sdf1".
# This might also look like "/dev/mmcblk0" (NOT "/dev/mmcblk0p1").
mmc=/dev/sdX
target_product_dir=out/target/product/$TARGET_PRODUCT
sudo linaro-android-media-create --mmc $mmc --dev panda --boot
$target_product_dir/boot.tar.bz2 --system $target_product_dir/system.tar.bz2
--userdata $target_product_dir/userdata.tar.bz2

./device/linaro/pandaboard/install-binaries-projectara.bash
```