Firefox OS

# Firefox OS build overview

This article is in need of a technical review.

**Building and installing Firefox OS requires a significant amount of time, network bandwidth, and computational power. Unfortunately, along the way, things are liable to go wrong. This page outlines the goals of the build process and the steps of that process in order to help users along the way. Details of each step are discussed in the linked pages.**

**Note:** The Firefox OS build process is full of references to 'B2G' or 'Build to Gecko'. 'Build to Gecko' was the original code name of the Firefox OS process.

## The Build Goal: four 'image' files

The overall purpose of the build process is to build four files that can be copied to a Firefox OS device.

| | |
|---|---|
| **boot.img** | The Linux kernel and a root filesystem image, the latter providing a usable set of basic Unix tools. |
| **system.img** | The core of Firefox OS including parts of Gonk, the port of Gecko, and the b2g executable. |
| **userdata.img** | The Gecko profile of the user and the Gaia web applications for the device. |
| **recovery.img** | A Linux kernel and a root filesystem image along with a simple tool to enable users to fix a bad installation. |

Once the four images have been created, they can be transferred to a device.

Firefox OS is built on top of the base Android Open Source Project (AOSP). The AOSP tools `adb` and `fastboot` provide powerful ways to access and manipulate a device. Notably, the command `adb reboot-bootloader` can cause a connected device to reboot and pause at the early bootloader stage where the command `fastboot flash $partition $image` can be used to copy an image onto the device.

## The Boot Image

The Boot Image (`boot.img`) is a combination of the Linux kernel and an initial root partition providing the core utility software and initialization script. The latter will be copied into device memory for efficient use by the device and therefore is called a "ramdisk". The Boot Image will be copied to the 'boot' partition on the device and the contents of the ramdisk are visible starting in the root directory when the device filesystem is accessed at runtime, such as when using `adb shell`.

The Boot Image also establishes the permissions of the root user in the `default.prop` file in the root directory.

It is also possible to modify existing boot images by inspecting the file, splitting the file into the kernel and ramdisk image, extracting the contents of the ramdisk image, modifying those contents, re-assembling the ramdisk image, then rebuilding a functional boot.img. See, for example, the Alcatel One Touch Fire Hacking (Mini) Guide page.

Boot Images can be tested before being installed by 'sideloading' them; the device can be started and paused in the bootloader and then `fastboot` can be used to boot from the Boot Image without installing it using the command `fastboot boot /some/path /to/boot.img`.

## The System Image

The System Image (`system.img`) provides the core of Firefox OS:

- **Gonk**: the low-level components of the operating system
- **Gecko**: the port of the Firefox HTML display and JavaScript engine
- **B2G**: the core runtime process of the operating system.

See the Firefox OS platform guide for more information about the platform architecture.

The System Image will be copied to the `system` partition on the device and will be visible in the `/system/` directory when the device filesystem is accessed at runtime.

**Note**: The System Image also provides the binary blobs that may be used by the device, notably the RIL (Radio Interface Layer) blob controlling the cellular radio on the device.

### The User Data Image

The User Data Image (`userdata.img`) provides the Gaia applications loaded at runtime.

The User Data Image will be copied to the `userdata` partition on the device and the contents will be visible in the `/data/` directory when the device filesystem is accessed at runtime. Notably the `/data/b2g/` directory contains the Mozilla Gecko *profile* of the device user while the `/data/local/webapps/` directory contains the actual web applications available to the user.

### The Recovery Image

The Recovery Image (`recovery.img`) contains the same kernel and a similar ramdisk as are present on the Boot Image partition. The recovery image however uses a different initialization script, which leads the user to a set of recovery commands accessible using the hardware buttons on the device.

The Recovery Image will be copied to the `recovery` partition on the device, which is not mounted onto the filesystem at regular runtime.

## The Build Process: setup, configure, build, install

The overall process of building and installing Firefox OS involves four steps:

| | |
|---|---|
| **Setup** | Obtain copies of all the programs used by the build process, such as the right compilers and libraries. |
| **Configure** | Download the source code that will be built and create the `.configure` file that defines environmental variables specifying the paths and other values used in the build. |
| **Build** | Build the Gecko profile of the user and the Gaia web applciations for the device. |
| **Install** | Install the files on a device. |

## Setup

Inital setup must be done to ensure the computer running the build has all of the software required during the build, such as compilers and build tools.

This step can be done by hand or using a script. Details are discussed in the Firefox OS build prerequisites page.

**Note**: On UNIX and UNIX-like machines, the presence of the required software can be checked using the unix command `which` with the name of the required program as a parameter.

## Configuration

The actual build process starts with obtaining a copy of the Firefox OS (or B2G) software, usually by creating a Git clone of the `B2G` project. The build configuration will both obtain copies of all the source code which is to be built and create the `.config` file that specifies variables for the build.

This is run with the `config.sh` script. Details are discussed in the Preparing for your first B2G build page.

The configure script needs a parameter specifying the type of device to build. The build names are code names linked to the CPU architecture rather than a specific device, and there is currently no way to establish which build works for which physical device. A list of available code names can be found here.

The configure step will also use the Android Open Source Project `repo` tool to download (or update) a copy of all the code used in the build. These copies will be stored in the `.repo/projects` directory. Due to this activity, the configure step can take a long time and will download a lot of data.

## Build

The build step will actually compile all of the source code and produce the output images.

This is run with the `build.sh` script. Details are discussed in the Building Firefox OS page.

By default, the build step is monolithic, attempting to build everything at once from the Android Open Source Project tools to the Linux kernel to the Gaia web applications. When the build fails, it can sometimes be unclear in which step it has failed.

It is possible to build only certain parts of the whole Firefox stack. For example, the Gecko system only can be built by calling the build script with the `gecko` parameter. Similarly, Gaia can be built on its own using the `gaia` parameter. These parts can then be installed separately onto the device as explained next.

It is also possible to build the images discussed in the first part of this page. For example, the system image can be built using `./build.sh out/platform/$target/system.img`, where the `$target` parameter is the same as given in the Configuration step.

## Install

The install step will place the newly compiled code onto a device. This is run with the `flash.sh` script.

Individual parts of the build can be installed by adding a parameter to the flash script. For example, it is possible to install only the gaia web applications by specifying `./flash.sh gaia`.