Preparing Devices for Automation

The following guides will help to prepare hardware devices for automation.

For Android rooted devices that use Magisk keep in mind that currently we can't use Magisk version higher than 17.3
See bug: https://bugzilla.mozilla.org/show_bug.cgi?id=152540

Google Pixel 8

Written by: animalan@mozilla.com (animalan on Slack)

Install Android Platform Tools
Install using `./mach bootstrap` (select GeckoView/Firefox for Android, install adb)
or `brew install android-platform-tools` (installs adb, fastboot)

Enable Settings for Rooting
Go to Settings > About Phone
Tap Build Number 7 times to enable Developer Options
Go to Settings > System > Developer Options
Enable USB Debugging
Enable OEM Unlocking
Turn off Automatic System Updates

Unlock the Bootloader
Restart/reboot phone and hold volume down and power button to enter fastboot mode
Connect the phone to pc and run fastboot devices to verify connection (device serial number should be displayed)
Use phone's volume buttons to select "Unlock bootloader", press power button to confirm
Device state should change from locked to unlocked
Check Enable OEM Unlocking settings in the Developer Options. It should display the bootloader is unlocked.

Patch init_boot.img
Install Magisk v29.0 on the Pixel 8 (As of May 20 2025, the Magisk 17.3 problem mentioned above appears to be resolved …?)
Download the factory image that matches the phone's build number (found in Settings > About your Phone): (e.g. AP1A.240305.019.A1, March 2024)
Unzip the factory image and locate `init_boot.img`
Tap Install > Select and Patch a File
Choose `init_boot.img`
Magisk will output a patched file (e.g. `magisk_patched-29000_liuWA.img`)

Install patch
Reboot the phone into Fastboot mode and connect the phone to a PC
Run `fastboot flash init_boot magisk_patched-29000_liuWA.img`
Reboot the phone

Verification
Open Magisk to verify root (installed version 29.0 should be displayed)
Connect device to PC
Ensure that [SharedUID] Shell in the Superuser tab is turned on. This enables Superuser rights (su) root user.
Run the following commands and check for the following output to verify root access:

```
> adb shell
> shiba:/ $ whoami
shell
> shiba:/ $ su
> shiba:/ # whoami
root
> shiba:/ # su -v
29.0:MAGISKSU # Root access provided by MAGISK
```

Rooting Completed.

Google Pixel 6

Steps 1a to 1i are the same on the Pixel6 as they were on the Pixel2

- These steps are preparing adb on your computer and phone and are the same across both devices(and alxl of android I think)
- I used this guide and youtube video to root the Pixel6:
    - https://droidwin.com/root-google-pixel-6-via-magisk/
    - https://www.youtube.com/watch?v=hdlF6ZdLwDU

1.

- j)
    - Go to https://developers.google.com/android/images and download the factory image for your build number (found above) and matching type (walleye or taimen). For the pixel6 you must download the image for the Pixel7 equivalent, **yes that is written correctly, the Pixel7 image for the Pixel6**
- k)
    - After downloading, unzip it, and unzip the zip (starting with **image***) found inside
- l)
    - In that folder you will find a file called **boot.img**
- m)
    - Run: **adb push <PATH/TO>/boot.img /sdcard/Download/**

- n)
    - Confirm that it exists now: **adb shell ls /sdcard/Download**

2.

1. Download Magisk 17.3:
    a. Repo: https://github.com/topjohnwu/Magisk/releases/tag/v25.2
2. Open the Magisk app, click install
3. Click select and patch a file
4. Select the stock boot img file from m)
5. Should take a few seconds to complete
6. You should see:
   Output file is written to:
   /storage/emulated/0/Downloads/magisk_patched-***.
7. Copy and place that file somewhere on your computer

3.

RUN: adb reboot bootloader

Wait for the phone to enter the fastboot mode

4.

Run: fastboot boot path/to/magisk_patched-.img (The .img file from step 2.7)**

Not recommended at all but you can do flash boot, rather than fastboot boot but this is not recommended as flashing makes it permanent, fastboot boot changes can entirely avoid bootloops as all changes will be reverted on a restart. Better to do it using fastboot incase any mistakes were made as **flashing cannot be undone**

5.

1. Open the magisk app
2. Select install within the application
3. Click “Direct Install(Recommended)”
4. Click “Let’s go”
5. Wait a few seconds for steps to complete and click “reboot”
6. Now if anything goes wrong as long as you used the fastboot boot command all changes will be reverted upon a subsequent reboot

6.

Download a root checker app to check you have root access(sample app icon below)



Google Pixel 2

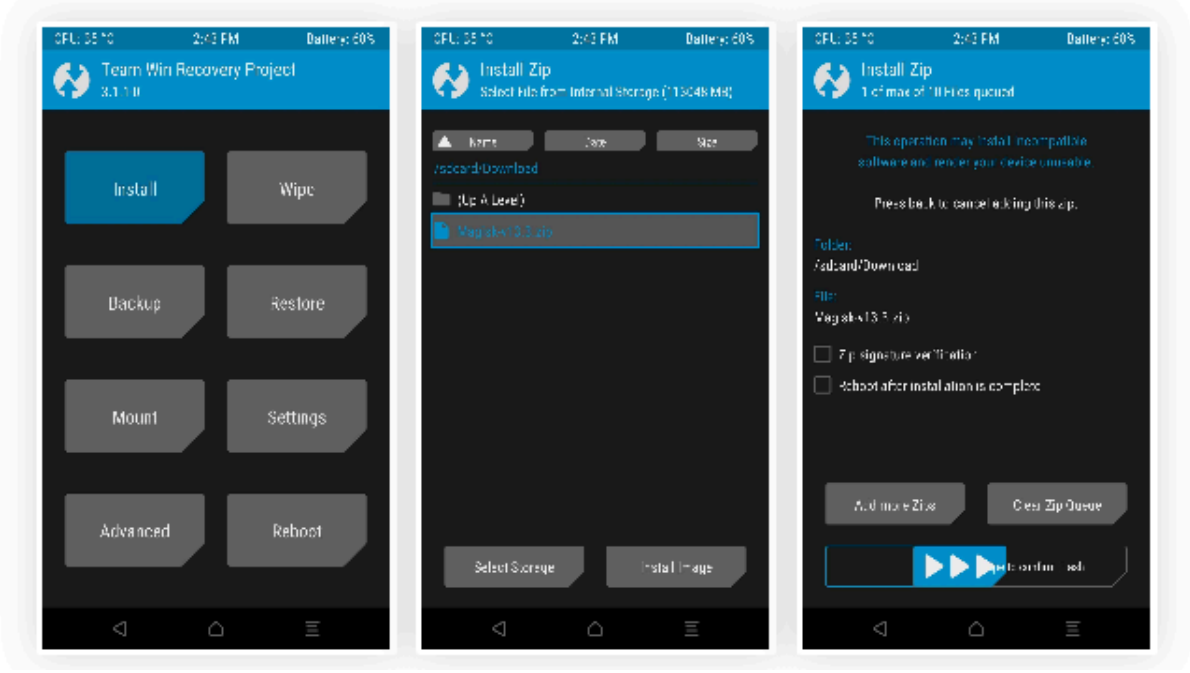
Some details are here:
https://gist.github.com/rwood-moz/b2c1db3a9820a0a3ff74fb662407fd97
su
As for the bug above (1525401) and because magisk manager won't install an older version of Magisk here are some updated steps:

1. Unlock the bootloader (taken and modified from here):
    a. **Download and install ADB**: First make sure you have the Android SDK/SDK Tools installed on your local machine. Attach your android device to USB, then in a terminal type **adb devices** and see if your device is listed. If ADB is not found locally, you can install it by running **./mach bootstrap** in your mozilla repo and selecting 'Firefox for Android Artifact Mode' and bootstrap will then install the required tools (no need to actually build).
        i. Make sure **adb** is in the path (required for running raptor).
        ii. If running from Ubuntu, be sure to have the USB mode set to **Transfer Files**.
        iii. On Windows, you can also follow these instructions: https://www.xda-developers.com/install-adb-windows-macos-linux/
    b. **Enable Developer Mode**: Tap the **Build Number** option in the **settings -> system -> about** screen several times. A popup will be displayed saying 'you are now a developer'. Now when you go to **settings -> system** menu a **Developer Options** item is there.
    c. In the **Settings -> System -> Developer Options** screen, turn on **USB debugging**.
        i. For any platform, make sure the toggle **Verify apps over USB** is set to off.
    d. Test the connection with the command **adb devices** and you should see the phone with device next to it.
        i. If you see 'unauthorized' instead of device, open the phone and allow the computer to connect.
        ii. If you have a 'permission' denied error, check if the USB connection is set to **Transfer Files**.
    e. If the device is already rooted the following commands will work, and won't throw an error:
        i. **adb shell**
        ii. **su**
        iii. **ls /data/data**
        iv. Another one to try (not necessary if the others failed with permission denied errors): **sudo ls /data**
    f. Next in Developer Options turn ON **OEM Unlocking** and turn OFF **Automatic System Updates**, save.
        i. If **OEM Unlocking** is greyed, do the following (be careful here!):

1. Run: **fastboot flashing lock_critical**
2. Select the **unlock** option
3. It'll factory reset now - go to step **h** once complete.

g. Unlock bootloader now with (for Moto G5 Cedric follow the instructions below):
   i. **adb reboot bootloader**
   ii. **fastboot flashing unlock**
      1. **This is done by holding down the volume down and the power button**
   iii. **fastboot reboot**
   iv. After the last command, the phone will perform a factory reset.
h. Find the **Build Number** by going through **Settings -> About Phone**, or **System -> About Phone** for older android versions). i.e. OPM2.171019.029
i. Determine the phone type by running **adb shell** (you should see either **walleye** or **taimen** - **sailfish** is another version) and take note of it.
j. Go to https://developers.google.com/android/images and download the factory image for your build number (found above) and matching type (walleye or taimen).
   i. If thereafor Verizon) you'll know you have the Verizon version if "OEM Unlocking" was greyed out (from earlier).
   ii. For Moto G5, use one of the stock images found here: https://forum.xda-developers.com/g5/development/official-stock-moto-g5-cedric-firmware-t3733897
k. After downloading, unzip it, and unzip the zip (starting with **image***) found inside (this only applies to Pixel 2, for Moto G5 you will find **boot.img** after the first extraction).
l. In that folder you will find a file called **boot.img**
m. Run: **adb push <PATH/TO>/boot.img /sdcard/Download/**
n. Confirm that it exists now: **adb shell ls /sdcard/Download**

2. Download TWRP:
   a. For Pixel2: https://twrp.me/google/googlepixel2.html
      i. For OPD3.170816.012 Android 8.0.0 walleye, this download is known to work twrp-3.2.3-1-walleye.img
   b. For Moto G5 Cedric: https://dl.twrp.me/cedric/twrp-3.2.1-0-cedric.img.html
3. Download Magisk 17.3:
   a. Repo: https://github.com/topjohnwu/Magisk/releases/tag/v17.3
   b. File to download: https://github.com/topjohnwu/Magisk/releases/download/v17.3/Magisk-v17.3.zip
   c. For Pixel2 17.3 did not work, but 17.2 did work, using a version higher than 17.3 will cause a soft-brick
4. **adb reboot bootloader**
5. **fastboot boot path/to/twrp.img**
6. **adb push <path to magisk 17.3 .zip> /sdcard/Download**
7. Follow these next steps on the device (a TWRP screen may be asking for continuing without read access on the device, make it read-only):

a. **Tap install**
b. **Use file manager to open /sdcard/Download**
c. **Select Magisk-v17.3.zip**
d. **Swipe to install**
e. **Tap reboot**
f. **Select don't install TWRP app and reboot**
g. 

8. After reboot open Magisk and check if reboot works, then check the following commands:
    a. **adb shell**
    b. **su**
    Allow the request in Magisk Manager (you may have to type su a second time to become superuser after accepting the request). Or set **Automatic Response** to grant in the Magisk Manager settings. After the first request, a slider will become available in the **# Superuser** menu.
    c. **ls /data/data**
9. Run raptor test to check if everything is OK
    a. Before starting, in the phones Developer Options menu, toggle the **Stay Awake** option to on. (Tests will fail if the screen isn't on when raptor starts).
    b. Download **geckoview_example.apk** from latest Android 4.0 API16+ opt build on central
    c. Install with: **adb install -g <PATH/TO>geckoview_example.apk**
    d. If you haven't already, run **./mach build** (artifact mode is fine)
    e. Then run the tests with: **./mach raptor-test --test raptor-speedometer --app=geckoview --binary=org.mozilla.geckoview_example**
10. For Moto G5 and Pixel 2, you may need to install the WakeLock app from the PlayStore (make sure you disable auto-updates in the play store when you login). Once installed, set it to the 6th setting (the one before Full Lock).
11. For Moto G5, an intermittent screen needs to be disabled now:
    a. Go to **Settings -> Apps**

b. Click the three dots in the upper-right corner and select **Show System**.
c. Scroll down to **Motorola Update Services** and click it.
d. Select **Notifications** and toggle **Block All** to On.

Motorola G5

Same steps as on Pixel2 but make sure you use the correct TWRP
https://twrp.me/motorola/motorolamotog5.html

To unlock the bootloader on Motorola G5 Plus Cedric, do the following:

1. Go to this website. Read the information given if you wish and click on Next button to proceed to the next step.
2. Sign into a google or motorola account here.
3. Follow their instructions and go the Command Prompt and type fastboot oem get_unlock_data.
4. A new serial code will appear in your command prompt screen and you have to copy the entire code (multiple lines, delete the new lines and (bootloader) prefixes) and enter this code into the motorola web page.
5. Move down to the bottom of the page after checking if the phone can be unlocked and select I Agree and then Request an Unlock Key.
6. You will receive an unlock key through the email you provided for the Google/Motorola sign-in.
7. Now, run the following in the command prompt (you might need to run it twice): **fastboot oem unlock <OEM_UNLOCK_KEY>**
8. The device might reset now, if it doesn't run **fastboot reboot**.
9. It should reformat the phone now. Ignore the **ID: bad key** error, it is nothing to worry about.
10. The device is now unlocked!

Amazon Fire TV

You must enable both adb and debugging on your Fire TV device before you can connect to it:

1. From the main screen of your Fire TV, select **Settings**.
2. Select **My Fire TV > Developer Options**.
3. Turn on **ADB Debugging**.
4. Turn on **Apps from Unknown Source**s.
5. Turn off the screen saver by setting a custom value of 24 days:
    a. **adb shell settings put system screen_off_timeout 2147460000**
6. Turn off sleep timer by setting a custom value of 24 days
    a. **adb shell settings put secure sleep_timeout 2147460000**
7. Run raptor test to check if everything is OK
    a. Download geckoview_example.apk from latest Android 4.0 API16+ opt build on central

    b. Install: **adb install -g <PATH/TO>geckoview_example.apk**
    c. Run: **./mach raptor-test --test raptor-speedometer --app=geckoview --binary="org.mozilla.geckoview_example"**

For detailed streaming information some overlays can be enabled via system x-ray:

1. Run **adb shell am start com.amazon.ssm/com.amazon.ssm.ControlPanel**
2. Enable "System X-Ray" and "Advanced Options"

Lenovo Yoga

Just follow the default setup steps to prepare your device for your own use. Afterward do these steps:

1. Switch out of S Mode to be able to install custom applications
2. Setup Mozilla Build environment

In case you have to recover the OS you have different options:

1. Reset the device via the internal OS partition: https://support.lenovo.com/th/nn/solutions/ht502594
2. Create a recovery USB key: https://support.lenovo.com/gb/en/solutions/ht103653

Note: The required Serial Number can be found at the bottom of the device

Samsung A51 (Android 11)

See here for rooting instructions: A51 Rooting/Setup Guide

Samsung A55 (Android 14)

See here for rooting instructions: A55 Rooting/Setup Guide