

Firmware Update Types

RW_LEGACY

- * Updates/replaces the stock legacy boot payload (SeaBIOS) included on many models; supplements the ChromeOS / secure boot payload
- * Leaves all stock functionality intact, including the Developer Mode boot screen and Recovery Mode functionality
- * Allows device to dual-boot another OS alongside ChromeOS
- * Does not require disabling the firmware write-protect
- * Carries zero risk of bricking the device
- * On many models, includes bugfixes and enables additional functionality

RW_LEGACY firmware is for users who want to dual-boot ChromeOS + Linux, or users who want to run Linux without having to open the device/disable the firmware write-protect (and are ok with the accompanying limitations).

BOOT_STUB (now deprecated; listed for historical purposes only)

- * Replaces the ChromeOS Verified Boot firmware payload (depthcharge) with a legacy boot payload (SeaBIOS)
- * Removes the Developer Mode boot screen (white "OS verification is OFF")
- * Removes the ability to run ChromeOS
- * Requires the firmware write-protect to be disabled
- * When flashed via the ChromeOS Firmware Utility Script, has a "built-in" backup function, so easy to restore without having to keep track of the backup file
- * Leaves the Recovery Mode (\[ESC+REFRESH+POWER\]) initiation intact
 - * If you engage this key sequence, your device will try to boot into Recovery Mode, which no longer exists
 - * To escape the Recovery Mode "black hole", you will have to disassemble the device and briefly disconnect the battery
 - * TL;DR - be careful not to do it :)
- * Was a stop-gap solution for Baytrail devices, which for a long time didn't have a Full ROM firmware available

BOOT_STUB firmware was originally an option for some platforms which lacked UEFI (Full ROM) firmware support, but it's no longer needed and support for it has been dropped).

(UEFI) Full ROM

- * A complete firmware image which includes updated/customized versions of the hardware init component (coreboot) and UEFI boot payload (edk2/Tianocore)
- * Removes the developer mode boot (white "OS verification is OFF") screen
- * Completely removes the ability to run ChromeOS (and ChromeOS Recovery Mode), creating a small risk of bricking your device
- * Offers the best support for booting all OSes besides ChromeOS
- * Adds full hardware support for virtualization (vmx / VT-x)
- * Fixes many bugs and/or idiosyncrasies associated with the stock firmware
- * UEFI firmware contains updated EC firmware as well, which brings additional fixes on most Chromebooks (mainly keyboard related)

- * Requires installation of a UEFI-compatible OS after flashing
- * Essentially turns your ChromeOS device into a "regular" PC / laptop

The **(UEFI) Full ROM** firmware is the best option for all users who no longer need/want to run ChromeOS (ie, want to run Linux/Windows exclusively), and who don't mind opening their device to disable the firmware write-protect.

Firmware updates are exclusively available via the [ChromeOS Firmware Utility Script](#), though not all types are available for all devices. See the [Device Compatibility list](#) for full details.

ChromeOS Firmware Boot Flags (GBB Flags)

The ChromeOS firmware boot flags / Google Binary Block (GBB) flags are firmware level settings stored directly in the firmware flash chip itself in a read-only (RO) area, and therefore require the firmware write protect to be disabled before setting. The GBB flags control the behavior of the firmware payload used to boot ChromeOS (depthcharge); they are used to set the default boot mode (ChromeOS or Legacy Boot Mode), the Developer Mode boot screen timeout (2s or 30s), prevent exiting Developer Mode via spacebar, and to enable Legacy Boot Mode regardless of crosssystem dev_boot_legacy setting, among other things.

Important: The GBB Flags are only valid when using the the stock firmware, or stock + RW_LEGACY firmware update. They have no effect (and cannot be set) when using the UEFI Full ROM firmware.

Most users should not attempt to set these manually, but instead use the [ChromeOS Firmware Utility Script](#) to set them based on their preferred default boot mode and boot timeout (the script automatically sets the Developer Mode and Legacy Boot Mode overrides for all options except factory default). For those curious, a full description of all the GBB flags can be found in [Google's source code here](#).

Custom Firmware / Firmware Updates

The components of the stock firmware for any given ChromeOS device can be found in the device-specific branches of Google's firmware trees:

https://chromium.googlesource.com/chromiumos/third_party/coreboot

<https://chromium.googlesource.com/chromiumos/platform/depthcharge>

https://chromium.googlesource.com/chromiumos/third_party/seabios

For instance, the coreboot component for the Asus Chromebox CN60 (PANTHER) is

https://chromium.googlesource.com/chromiumos/third_party/coreboot/+/firmware-panther-4920.24.B

The Google firmware branch for a specific device is pretty much a fixed snapshot of the firmware component at/around the time of the device's release, and usually has minimal bugfix updates after release. In contrast, the firmware updates available here via the Firmware Utility Script are built using the latest upstream (main / master copy) source code directly from the coreboot and SeaBIOS projects, with additional fixes/tweaks applied on top.

The upstream repositories for coreboot and SeaBIOS can be found on github:

<https://github.com/coreboot/coreboot>

<https://github.com/coreboot/seabios>

My personal coreboot, SeaBIOS, Tianocore, and EC firmware repositories, along with the source for the scripts on this site, are also available on github:

<https://github.com/MrChromebox/coreboot>

<https://github.com/MrChromebox/SeaBIOS>

<https://github.com/MrChromebox/edk2>

<https://github.com/MrChromebox/scripts>

These repos contain not only the source used to build the current firmware releases, but also the build scripts and configurations used as well.

See [Compiling Your Own Firmware](#) for full info.