Chrome OS RMA shim

Contents

Overview

RMA stands for Return Merchandise Authorization. When there's a problem that the end user cannot solve, the user returns the device to the partner's service center for diagnosis and repair. The service center may swap components and reinstall the firmware and/or software image. For Chromebooks, that means the service center may need to disable write protection and change the HWID to match the new configuration.

RMA shim

Chromebooks are highly secured. With verified boot and write protection, it's difficult for the service center to run diagnosis and repair programs (usually built and customized by partners) because those won't be

signed by Google. Service centers may also have limited (or even no) network access. In general, what the partner needs is a tool that fulfills these requirements:

- The tool is signed by Google. An operator can boot the device by turning on the developer mode, attaching a USB stick and invoking recovery mode.
- The tool can run the partner's customized tool programs to check and verify components, very similar to the way the factory process works.

The RMA shim image is designed to meet these requirements. An RMA shim image is a combination of existing Chrome OS factory bundle components, all combined into one disk, including:

- Factory install shim
- Release image (FSI)
- Test image
- Factory toolkit
- HWID bundle
- Other optional components (firmware, complete script, etc.)

Universal RMA shim (Multi-board RMA shim)

A problem for regular (single-board) RMA shims is that we have to create separate per-board RMA shims for each project, which makes it hard to manage shim images and physical USB drives. A universal shim contains multiple RMA shims for different boards, which is easier to manage and distribute.

Pros:

- Reduce the number of shims to manage.

Cons:

- The size of a universal shim can be large. Each board in a shim takes about 3 GB, so a universal shim containing 3 boards will have size 9~10 GB.

Get the tool

image_tool is a useful tool to manage RMA shims. We can get this tool by downloading the factory public repo.

```
$ git clone https://chromium.googlesource.com/chromiumos/platform/factory
$ cd factory/
```

The tool is located at `setup/image_tool` . It's recommended to sync the git repo periodically to get the latest version.

```
(in factory/ repository)
$ git pull
```

After downloading the factory repo, we can run the unit test for RMA commands to check if it runs normally on the machine. The tool should be able to run in a fresh Linux environment without chroot.

```
$ py/tools/image_tool_rma_unittest.py
```

Create an RMA shim

To create an RMA shim, you should first get a factory bundle and follow the steps below.

Adjust RMA test list in factory toolkit

RMA test list is different from the test list used in factory manufacture line. For instance, there is no factory server during RMA. Hence, we need another test list for RMA.

The recommended way is to create a test list that inherits `generic_rma.test_list.json`, which already takes care of general RMA settings such as disabling factory server and enabling `rma_mode`, and then add factory tests to `RMAFFT` group.

```
{
  "inherit": [
    "generic_rma.test_list"
  ],
  "label": "RMA Test List for <project>",
  "definitions": {
    "RMAFFT": {
      "subtests": [
        ...
        ...
      ]
    }
  }
}
```

- In general, run all factory tests (runin and fatp) in the service centers with reduced test cycles. For example, reduce the duration of the stress test from 4 hours to 10 minutes.
- Verify that all spare mainboards used in service centers complete SMT tests.
- Verify that all spare mainboards have a registration code that was burned into RW_VPD during the factory process before sending the boards to service centers.
- Discuss with the OEM to finalize test items for the RMA process.
- **Do not** modify or remove any GRT (Google Required Test) items.
- Make sure the firmware write protection is enabled (which should be true if `constants.phase` is set to PVT).

Combine factory bundle components into an RMA shim image

After getting all the bundle components ready, we can combine these components into a single RMA shim image. To create an RMA shim image from a factory bundle, use `image_tool rma create` command:

```
$ setup/image_tool rma create \
    --board BOARD \
    --factory_shim path/to/factory_install_shim.bin \
    --test_image path/to/chromiumos_test_image.bin \
    --toolkit path/to/install_factory_toolkit.run \
    --release_image path/to/chromiumos_image.bin \
    --hwid path/to/hwid_bundle.sh \
    --output rma_image.bin
```

The command can be simplified if all the components are put in their respective bundle directories (`release_image/` , `test_image/` , etc.):

```
$ setup/image_tool rma create \
    --board BOARD \
    --output rma_image.bin
```

We can also specify the active test list when creating the RMA shim, so that we don't need to modify `active_test_list.json` in factory toolkit.

```
$ setup/image_tool rma create \
    --board BOARD \
    --output rma_image.bin \
    --active_test_list rma_main
```

Use an RMA shim

Flash the `rma_image.bin` to a USB drive, boot it with developer switch enabled in recovery mode (see following steps), and then the device will boot from the RMA shim.

Note: The following instructions only work for a Google signed RMA shim. If you are using a developer signed RMA shim, the boot process is the same as booting from a test image.

Flash an image to USB drive

Use `dd` command to flash a shim image to a USB drive or SD card, replacing `/dev/sdX` with the name of the USB/SD device.

```
$ sudo dd if=rma_image.bin of=/dev/sdX bs=8M iflag=fullblock oflag=dsync
```

If you have a Chromium OS development environment, you can also use `cros flash` command in chroot.

```
$ cros flash usb:// rma_image.bin
```

Boot from RMA shim (clamshells / convertibles)

1. Enter recovery mode.
2. Press `CTRL + D` to turn on developer switch.
3. Press `ENTER` to confirm.
4. Enter recovery mode again (no need to wait for wiping).
5. Insert and boot from USB stick with `rma_image.bin` .

Boot from RMA shim (tablets / detachables)

1. Enter recovery mode.
2. Press `VOL_UP + VOL_DOWN` to show recovery menu.
3. Press `VOL_UP` or `VOL_DOWN` to move the cursor to “Confirm Disabling OS Verification”, and press `POWER` to select it.
4. Enter recovery mode again (no need to wait for wiping).
5. Insert and boot from USB stick with `rma_image.bin` .

See here for instructions to enter recovery mode.

RMA shim menu

The RMA shim has a menu that allows the user to select an action to perform, which is described in Factory Installer README. Moreover, if the RMA shim is created using `image_tool rma create` command, the tool adds a flag `RMA_AUTORUN=1` in `lsb-factory` file, which sets the default action of the menu depending on the cr50 version and hardware write protection status, such that:

1. If hardware write protection is enabled, set the default action to **(E) Reset Cr50**, also known as RSU (RMA Server Unlock) to disable hardware write protection and enter factory mode. After RMA reset, the device will reboot. The user should enter recovery mode and boot to shim again.
2. If hardware write protection is disabled, set the default action to **(I) install** to install payloads from USB. If hardware write protection is disabled by disconnecting the battery instead of doing RSU, the install script will also enable factory mode at the end of installation.

You can stop the default action and return to shim menu by pressing any key within 3 seconds when the console prompts “press any key to show menu instead”.

During installation, you can remove the RMA shim when the copy is complete (the text color changes from yellow to green). After the installation, the device will boot into the test image with factory toolkit. Run through the factory tests to complete the flow. The last test should wipe out the factory test image and enable the release image.

Create a universal RMA shim

We can use `image_tool rma merge` command to create a universal shim using multiple RMA shims.

```
$ setup/image_tool rma merge \
    -i soraka.bin scarlet.bin \
    -o universal.bin
```

To delete a previously generated output image, specify the `-f` option:

```
$ setup/image_tool rma merge \
    -i soraka.bin scarlet.bin \
    -o universal.bin -f
```

Update a universal RMA shim

`image_tool rma merge` supports merging universal shims. If there are duplicate boards, it will ask the user to select which one to use. It can be used to update a board in a universal shim using an updated single-board RMA shim.

```
$ setup/image_tool rma merge \
    -i universal.bin soraka_new.bin \
    -o universal_new.bin
Scanning 2 input image files...

Board soraka has more than one entry.
================================================================================
(1)
From universal.bin
board         : soraka
install_shim  : 10323.39.28
release_image : 10575.37.0 (Official Build) dev-channel soraka
test_image    : 10323.39.24 (Official Build) dev-channel soraka test
toolkit       : soraka Factory Toolkit 10323.39.24
firmware      : Google_Soraka.10431.32.0;Google_Soraka.10431.48.0
hwid          : None
complete      : None
toolkit_config: None
lsb_factory   : lsb_factory
================================================================================
(2)
From soraka_new.bin
board         : soraka
install_shim  : 10323.39.31
release_image : 10575.37.0 (Official Build) dev-channel soraka
test_image    : 10323.39.24 (Official Build) dev-channel soraka test
toolkit       : soraka Factory Toolkit 10323.39.24
firmware      : Google_Soraka.10431.32.0;Google_Soraka.10431.48.0
hwid          : None
```

```
complete      : None
toolkit_config: None
lsb_factory   : lsb_factory
================================================================================
Please select an option [1-2]:
```

Use a universal RMA shim

Using a universal RMA shim is exactly the same as using a normal single-board RMA shim. Flash the image to a USB drive and boot from it using the instructions mentioned above.

Other RMA commands

There are other `image_tool` commands that makes verifying and modifying RMA shims easier. For detailed description and usage, please use the `--help` argument of the commands. For instance:

```
$ setup/image_tool rma show --help
```

Print bundle components in an RMA shim

`image_tool rma show` command can print the component versions in an RMA shim.

```
$ setup/image_tool rma show -i soraka.bin
This RMA shim contains boards: soraka
================================================================================
board         : soraka
install_shim  : 10323.39.31
release_image : 10575.37.0 (Official Build) dev-channel soraka
test_image    : 10323.39.24 (Official Build) dev-channel soraka test
toolkit       : soraka Factory Toolkit 10323.39.24
firmware      : Google_Soraka.10431.32.0;Google_Soraka.10431.48.0
hwid          : None
complete      : None
toolkit_config: cb5b52296cd4fcb0418b6879c0acc32b
lsb_factory   : d2c9d6a7d32ee3b1279c2b0b27244727
================================================================================
```

This command also applies to universal RMA shim.

```
$ setup/image_tool rma show -i universal.bin
This RMA shim contains boards: soraka scarlet
================================================================================
board         : soraka
install_shim  : 10323.39.31
```

```
release_image : 10575.37.0 (Official Build) dev-channel soraka
test_image    : 10323.39.24 (Official Build) dev-channel soraka test
toolkit       : soraka Factory Toolkit 10323.39.24
firmware      : Google_Soraka.10431.32.0;Google_Soraka.10431.48.0
hwid          : None
complete      : None
toolkit_config: cb5b52296cd4fcb0418b6879c0acc32b
lsb_factory   : d2c9d6a7d32ee3b1279c2b0b27244727
================================================================================
board         : scarlet
install_shim  : 10211.68.0
release_image : 10575.67.0 (Official Build) stable-channel scarlet
test_image    : 10211.53.0 (Official Build) dev-channel scarlet test
toolkit       : scarlet Factory Toolkit 10211.53.0
firmware      : Google_Scarlet.10388.26.0
hwid          : None
complete      : None
toolkit_config: None
lsb_factory   : c82d4c1f831bf20d7cdc70138fe4ef72
================================================================================
```

Replace bundle components in an RMA shim

`image_tool rma replace` command can replace components in an RMA shim. For instance, to replace the HWID bundle in an RMA shim with a new one:

```
$ setup/image_tool rma replace -i rma_image.bin --hwid new_hwid_bundle.sh
```

If the RMA shim is a universal shim, argument `--board` is needed.

```
$ setup/image_tool rma replace -i universal.bin \
    --board soraka --hwid new_hwid_bundle.sh
```

This command supports replacing `release_image`, `test_image`, `toolkit`, `factory_shim`, `firmware`, `hwid`, `complete_script` and `toolkit_config`.

Extract a single-board RMA shim from a universal shim

`image_tool rma extract` command can extract a single-board RMA shim from a universal shim.

```
$ setup/image_tool rma extract -i universal.bin -o extract.bin
Scanning input image file...

Please select a board to extract.
================================================================================
(1)
```

```
board         : soraka
install_shim  : 10323.39.31
release_image : 10575.37.0 (Official Build) dev-channel soraka
test_image    : 10323.39.24 (Official Build) dev-channel soraka test
toolkit       : soraka Factory Toolkit 10323.39.24
firmware      : Google_Soraka.10431.32.0;Google_Soraka.10431.48.0
hwid          : None
complete      : None
toolkit_config: cb5b52296cd4fcb0418b6879c0acc32b
lsb_factory   : d2c9d6a7d32ee3b1279c2b0b27244727
===============================================================================
(2)
board         : scarlet
install_shim  : 10211.68.0
release_image : 10575.67.0 (Official Build) stable-channel scarlet
test_image    : 10211.53.0 (Official Build) dev-channel scarlet test
toolkit       : scarlet Factory Toolkit 10211.53.0
firmware      : Google_Scarlet.10388.26.0
hwid          : None
complete      : None
toolkit_config: None
lsb_factory   : c82d4c1f831bf20d7cdc70138fe4ef72
===============================================================================
Please select an option [1-2]:
```

Edit lsb-factory config in an RMA shim

`image_tool edit_lsb` command can modify `lsb-factory` config, such as `RMA_AUTORUN` flag.

```
$ setup/image_tool edit_lsb -i rma_image.bin

Current LSB config:
===============================================================================
CHROMEOS_AUSERVER=http://...
CHROMEOS_DEVSERVER=http://...
FACTORY_INSTALL=1
HTTP_SERVER_OVERRIDE=true
FACTORY_INSTALL_FROM_USB=1
RMA_AUTORUN=true
===============================================================================
(1) Modify board to install.
(2) Modify Chrome OS Factory Server address.
(3) Modify default action (will be overridden by RMA autorun).
(4) Enable/disable countdown before default action.
(5) Enable/disable complete prompt in RMA shim.
(6) Enable/disable autorun in RMA shim.
```

```
(7) Modify cutoff config in cros payload (only for old devices).
(8) Enable or disable qrcode when factory reset.
(w) Apply changes and exit.
(q) Quit without saving changes.
Please select an option [1-9, w, q]:
```

or

```
$ setup/image_tool edit_lsb -i universal.bin --board soraka
```

Flags	Description	Option to modify	release shim version
CHROMEOS_RELEASE_BOARD	For using board specific config. (might be overridden by the value in firmware)	(1)	-
CHROMEOS_AUSERVER	Chrome OS Factory Server address.	(2)	-
CHROMEOS_DEVSERVER	Not used.	(2)	-
FACTORY_INSTALL_DEFAULT_ACTION	The factory shim will execute the default action automatically if not interrupted by user.	(3)	-
FACTORY_INSTALL_ACTION_COUNTDOWN	Countdown before doing default action, the countdown is 3 seconds	(4)	12387
FACTORY_INSTALL_COMPLETE_PROMPT	Wait for ENTER after action **(I) Install** is completed.	(5)	11766
RMA_AUTORUN	The factory shim will set the default action to **(I) Install** or **(E) Perform RSU**, depending on HWWP status.	(6)	11394
CUTOFF_METHOD, CUTOFF_AC_STATE, CUTOFF_BATTERY_MIN_PERCENTAGE, CUTOFF_BATTERY_MAX_PERCENTAGE, CUTOFF_BATTERY_MIN_VOLTAGE, CUTOFF_BATTERY_MAX_VOLTAGE, SHOPFLOOR_URL	Deprecated.	(7)	-
DISPLAY_QRCODE	Display the information of the DUT as a qrcode, to increase the flexibility of	(8)	15448

Flags	Description	Option to modify	release shim version
	customized process of factory reset.		
DISPLAY_INFO	Support fields are: `hwid`, `serial_number`, `mlb_serial_number`, `wifi_mac0`, `service_tag`. For example: `hwid serial_number, wifi_mac0` will display hwid and serial_number in the first qrcode, and display wifi_mac0 in the second qrcode.	(8)	15448
NETBOOT_RAMFS	This flag is automatically set to `1` when using netboot firmware. The factory shim will set the default action to **(I) Install**.	N/A	-

Note:

- Please do not directly mount the stateful partition and modify `lsb-factory` file. The actual config is stored in cros payload, so the modifications in the file will be overwritten.
-
- The install shim also checks `/etc/lsb-factory` for flags that decides the default action of the shim menu (listed from high priority to low priority).
  - `NETBOOT_RAMFS=1`
  - `RMA_AUTORUN=true`
  - `DEFAULT_ACTION=<action>`

Edit toolkit config in an RMA shim.

`image_tool edit_toolkit_config` command can modify toolkit config, such as active test list and cutoff config (after version 12162.0.0) and the config of customized reset process.

```
$ setup/image_tool edit_toolkit_config -i rma_image.bin

Toolkit config:
================================================================================
{
  "cutoff": {
```

```
    "CUTOFF_BATTERY_MAX_PERCENTAGE": 90,
    "CUTOFF_BATTERY_MIN_PERCENTAGE": 60,
    "CUTOFF_METHOD": "battery_cutoff",
    "CUTOFF_AC_STATE": "remove_ac",
    "CONTINUE_KEY": "s",
    "QRCODE_INFO": "serial_number"
  },
  "active_test_list": {
    "id": "main_rma"
  }
}
================================================================================
(1) Modify active test list.
(2) Modify test list constants.
(3) Modify cutoff config.
(4) Enable or disable a confirmation before battery cutoff.
(5) Enable or disable qrcode right before cutoff.
(6) Modify the config to perform customized reset process.
(q) Quit without saving changes.
(w) Apply changes and exit.
Please select an option [1-3, q, w]:
```

or

```
$ setup/image_tool edit_toolkit_config -i universal.bin --board soraka
```

Flags	Description	Option to modify
active_test_list.id	The default test list when starting toolkit.	(1)
test_list_constants	The constant variables in test list.	(2)
cutoff	Cutoff process is at the end of factory reset. Check cutoff README for more information.	(3), (4), (5)
custom_reset_process	The config to perform customized reset process. Check CUSTOM_RESET_PROCESS for more information.	(6)

Unpack and repack toolkit in an RMA shim.

`image_tool payload toolkit` command can unpack and repack the factory toolkit in an RMA shim.

```
$ setup/image_tool payload toolkit -i rma_image.bin --unpack toolkit_path
(Edit some files in toolkit_path/ ...)
$ setup/image_tool payload toolkit -i rma_image.bin --repack toolkit_path
```

or

```
$ setup/image_tool payload toolkit \
    -i universal.bin --board soraka --unpack toolkit_path
(Edit some files in toolkit_path/ ...)
$ setup/image_tool payload toolkit \
    -i universal.bin --board soraka --repack toolkit_path
```

Powered by Gitiles| Privacy| Terms

source log blame