Installation of RedHawk™ 8.4.8 Gold on the NVIDIA® Jetson AGX Xavier

Release Notes

October 23rd, 2023

0898003-8.4.8-Xavier

READ ME BEFORE
INSTALLING
THIS PRODUCT

concurrent
REAL-TIME

This page intentionally left blank

1. Introduction

This document describes the process of installing the ARM64 version of RedHawk™ 8.4.8 on the Jetson AGX Xavier. The directions in this document supersede all others – they are specific to installing the software on Concurrent Real-Time's RedHawk systems.

2. Requirements

- Jetson AGX Xavier with Internet access
- An x86_64 host system running Ubuntu 20.04 with Internet access; note that the host system should be up to date with the latest available Ubuntu package updates
- The *RedHawk 8.4.8 for Jetson AGX Xavier* optical media disk

3. Installation

3.1. Download SDK Manager onto the host system

On an x86_64 host system running Ubuntu 20.04, download version 2.0.0-11402 of SDK Manager from the NVIDIA website via this URL:

https://developer.nvidia.com/jetpack

Click on the NVIDIA SDK Manager link to download the SDK Manager package (login may be required). You should also click on the Jetson AGX Xavier Developer Kit User Guide to download the User's Guide PDF.

Place these downloaded files into a `xavier` directory in the home directory of the current user.

3.2. Install the downloaded SDK Manager package

Open a terminal window and issue the following commands to install the SDK Manager:

```
$ cd ~/xavier
$ sudo apt install ./sdkmanager_2.0.0-11402_amd64.deb
```

3.3. Connect the Xavier system to the host system

The Xavier must be connected to the host system for SDK Manager to be able to detect it and successfully flash it. Two different USB cables are required to connect the Xavier to the host system. Refer to the cables and documentation included with your Xavier.

3.4. Put the Xavier into recovery mode

Newer versions of SDK Manager work best when the Xavier is initially placed into recovery mode. If the Xavier is powered on and booted, issue the following command to enter recovery mode:

```
reboot --force forced-recovery
```

Alternatively, connect the power cable with the middle button pressed, then press the left button, and then release both buttons at the same time. Issue `lsusb` on the host to verify that an `NVidia Corp.` entry with device ID `0955:7019` is displayed; this indicates the Xavier has entered recovery mode.

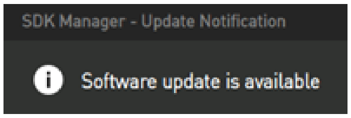
3.5. Use SDK Manager to flash the Xavier

Invoke the following command as the current user to begin using the SDK Manager:

```
$ sdkmanager
```

The first action required by the SDK Manager is to log into the nvidia.com website using your developer or partner account email address and password. Enter the appropriate information and press LOGIN to continue.

After logging in you may be presented with the following Update Notification dialog:



SDK Manager - Update Notification

Software update is available

If this dialog appears, **do not** click Update Now and instead click Later so that flashing can be performed with the currently installed SDK Manager, which should be at version 2.0.0-11402.

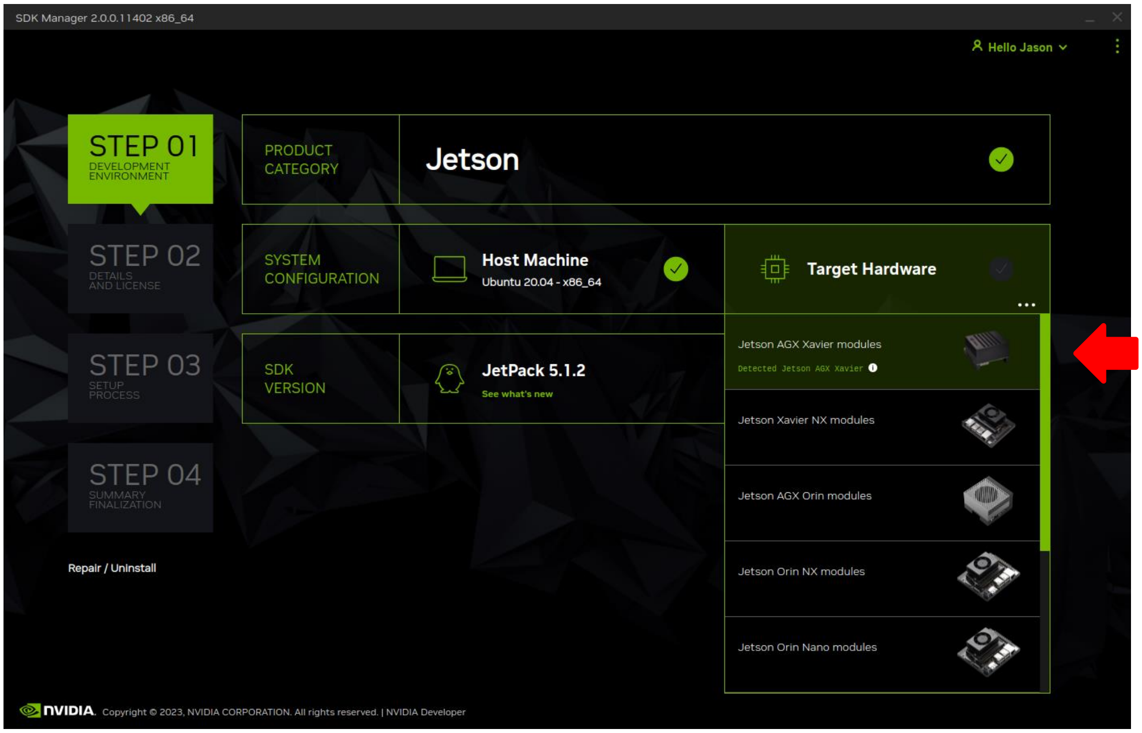
NOTE

RedHawk 8.4.8 has only been validated with JetPack version 5.1.2.
Other versions are incompatible with RedHawk 8.4.8 and must not be used.

At this point, you can generally follow the instructions in the *Jetson AGX Xavier Developer Kit User Guide* along with the following RedHawk-specific instructions.

SDK Manager screenshots are provided below.

1. On the STEP 01 - DEVELOPMENT ENVIRONMENT screen, click on the . . . icon in the Target Hardware box to reveal the list of supported hardware configurations, as shown in the following screenshot:



SDK Manager 2.0.0.11402 x86_64

Hello Jason

STEP 01 DEVELOPMENT ENVIRONMENT

STEP 02 DETAILS AND LICENSE

STEP 03 SETUP PROCESS

STEP 04 SUMMARY FINALIZATION

Repair / Uninstall

PRODUCT CATEGORY: Jetson

SYSTEM CONFIGURATION: Host Machine Ubuntu 20.04 - x86_64 | Target Hardware

SDK VERSION: JetPack 5.1.2 See what's new

Jetson AGX Xavier modules
Detected Jetson AGX Xavier

Jetson Xavier NX modules

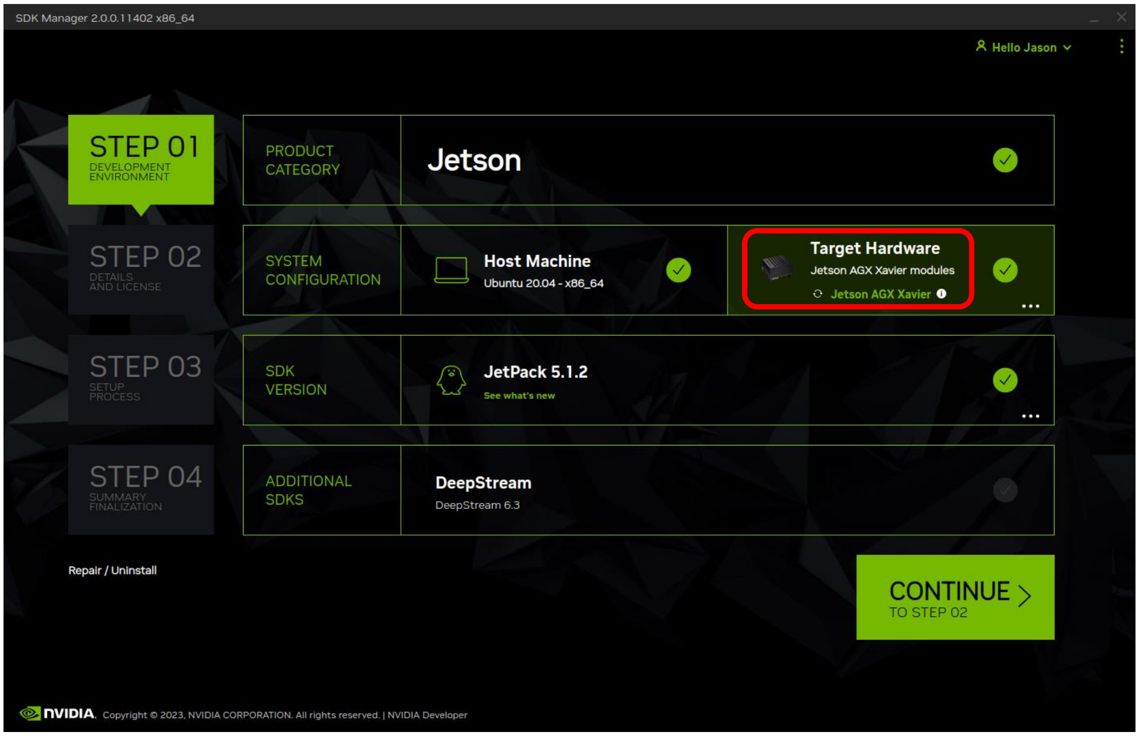
Jetson AGX Orin modules

Jetson Orin NX modules

Jetson Orin Nano modules

NVIDIA. Copyright © 2023, NVIDIA CORPORATION. All rights reserved. | NVIDIA Developer

Click on the Jetson AGX Xavier modules target hardware option to make it the selected target, as shown in the following screenshot:



SDK Manager 2.0.0.11402 x86_64

Hello Jason

STEP 01 DEVELOPMENT ENVIRONMENT

STEP 02 DETAILS AND LICENSE

STEP 03 SETUP PROCESS

STEP 04 SUMMARY FINALIZATION

PRODUCT CATEGORY: Jetson ✓

SYSTEM CONFIGURATION: Host Machine, Ubuntu 20.04 - x86_64 ✓ | Target Hardware, Jetson AGX Xavier modules, Jetson AGX Xavier ✓

SDK VERSION: JetPack 5.1.2 ✓ See what's new

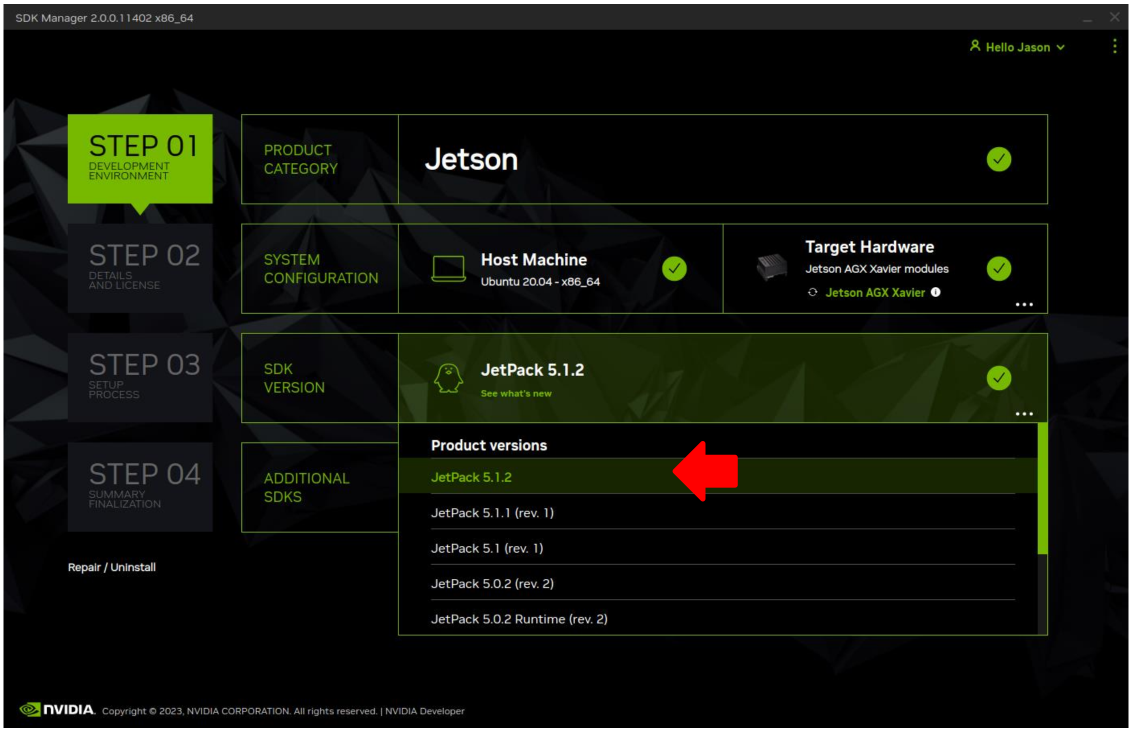
ADDITIONAL SDKS: DeepStream, DeepStream 6.3

Repair / Uninstall

CONTINUE TO STEP 02

NVIDIA. Copyright © 2023, NVIDIA CORPORATION. All rights reserved. | NVIDIA Developer

2. Click on the . . . in the Linux box to reveal the list of supported JetPack versions, as shown in the following screenshot:



SDK Manager 2.0.0.11402 x86_64

Hello Jason

STEP 01 DEVELOPMENT ENVIRONMENT

STEP 02 DETAILS AND LICENSE

STEP 03 SETUP PROCESS

STEP 04 SUMMARY FINALIZATION

PRODUCT CATEGORY: Jetson ✓

SYSTEM CONFIGURATION: Host Machine, Ubuntu 20.04 - x86_64 ✓ | Target Hardware, Jetson AGX Xavier modules, Jetson AGX Xavier ✓

SDK VERSION: JetPack 5.1.2 ✓ See what's new

ADDITIONAL SDKS

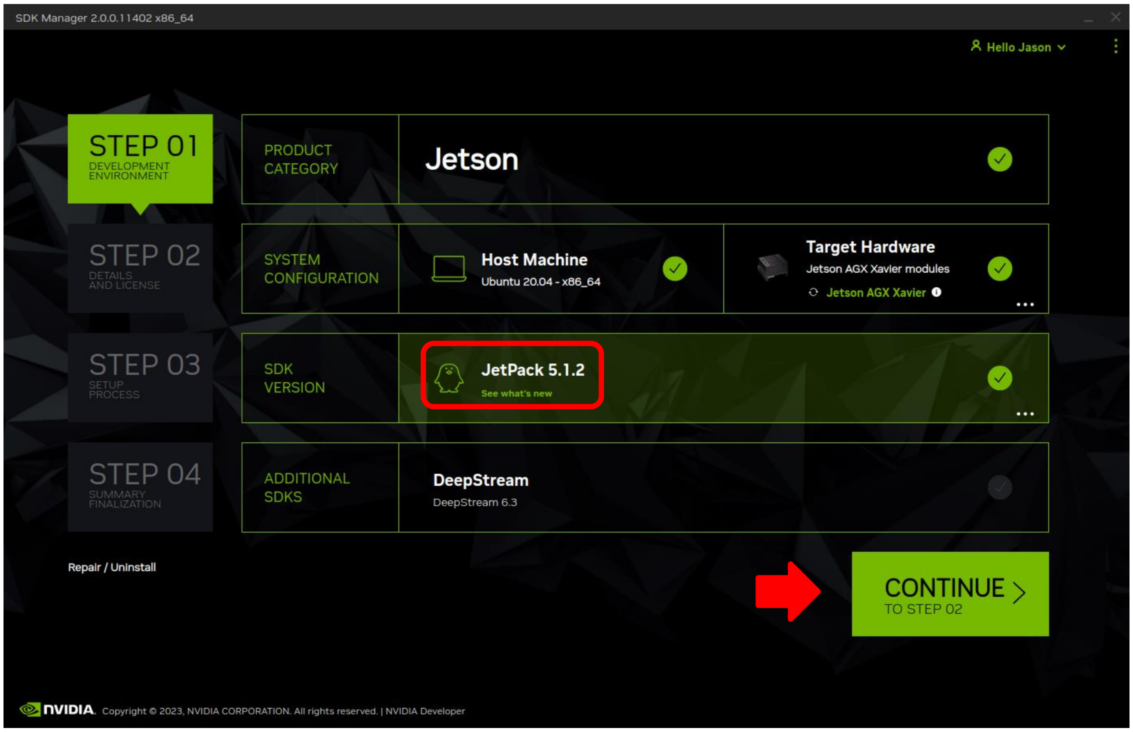
Product versions

- JetPack 5.1.2
- JetPack 5.1.1 (rev. 1)
- JetPack 5.1 (rev. 1)
- JetPack 5.0.2 (rev. 2)
- JetPack 5.0.2 Runtime (rev. 2)

Repair / Uninstall

NVIDIA. Copyright © 2023, NVIDIA CORPORATION. All rights reserved. | NVIDIA Developer

3. Click on the JetPack 5.1.2 version to make it the selected version, as shown in the following screenshot:


SDK Manager 2.0.0.11402 x86_64
Hello Jason
STEP 01 DEVELOPMENT ENVIRONMENT
STEP 02 DETAILS AND LICENSE
STEP 03 SETUP PROCESS
STEP 04 SUMMARY FINALIZATION
Repair / Uninstall
CONTINUE TO STEP 02
NVIDIA. Copyright © 2023, NVIDIA CORPORATION. All rights reserved. | NVIDIA Developer

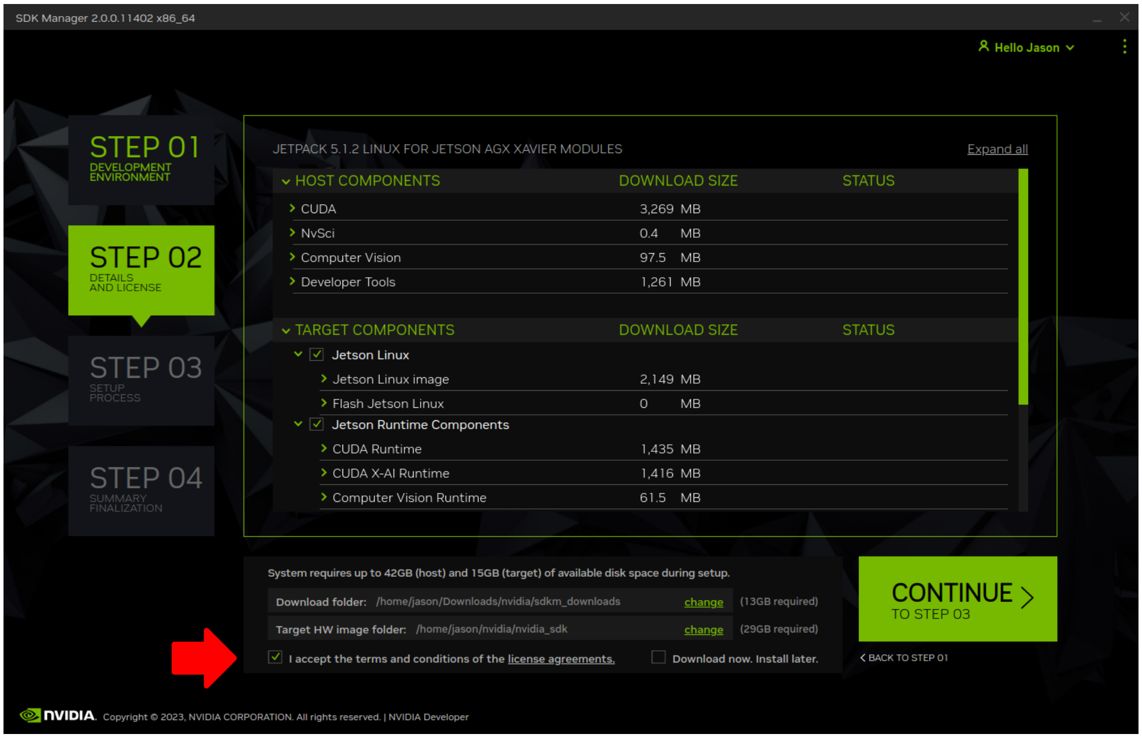
PRODUCT CATEGORY: Jetson

SYSTEM CONFIGURATION: Host Machine — Ubuntu 20.04 - x86_64; Target Hardware — Jetson AGX Xavier modules — Jetson AGX Xavier

SDK VERSION: JetPack 5.1.2 — See what's new

ADDITIONAL SDKS: DeepStream — DeepStream 6.3

4. Press Continue to advance to the STEP 02 – DETAILS AND LICENSE screen and click I accept the terms and conditions of the license agreements at the bottom of the window to accept the licenses, as shown in the following screenshot:


SDK Manager 2.0.0.11402 x86_64
Hello Jason
STEP 01 DEVELOPMENT ENVIRONMENT
STEP 02 DETAILS AND LICENSE
STEP 03 SETUP PROCESS
STEP 04 SUMMARY FINALIZATION
Expand all
CONTINUE TO STEP 03
BACK TO STEP 01
NVIDIA. Copyright © 2023, NVIDIA CORPORATION. All rights reserved. | NVIDIA Developer

JETPACK 5.1.2 LINUX FOR JETSON AGX XAVIER MODULES

HOST COMPONENTS	DOWNLOAD SIZE	STATUS
CUDA	3,269 MB	
NvSci	0.4 MB	
Computer Vision	97.5 MB	
Developer Tools	1,261 MB	
TARGET COMPONENTS	**DOWNLOAD SIZE**	**STATUS**
☑ Jetson Linux		
Jetson Linux image	2,149 MB	
Flash Jetson Linux	0 MB	
☑ Jetson Runtime Components		
CUDA Runtime	1,435 MB	
CUDA X-AI Runtime	1,416 MB	
Computer Vision Runtime	61.5 MB	

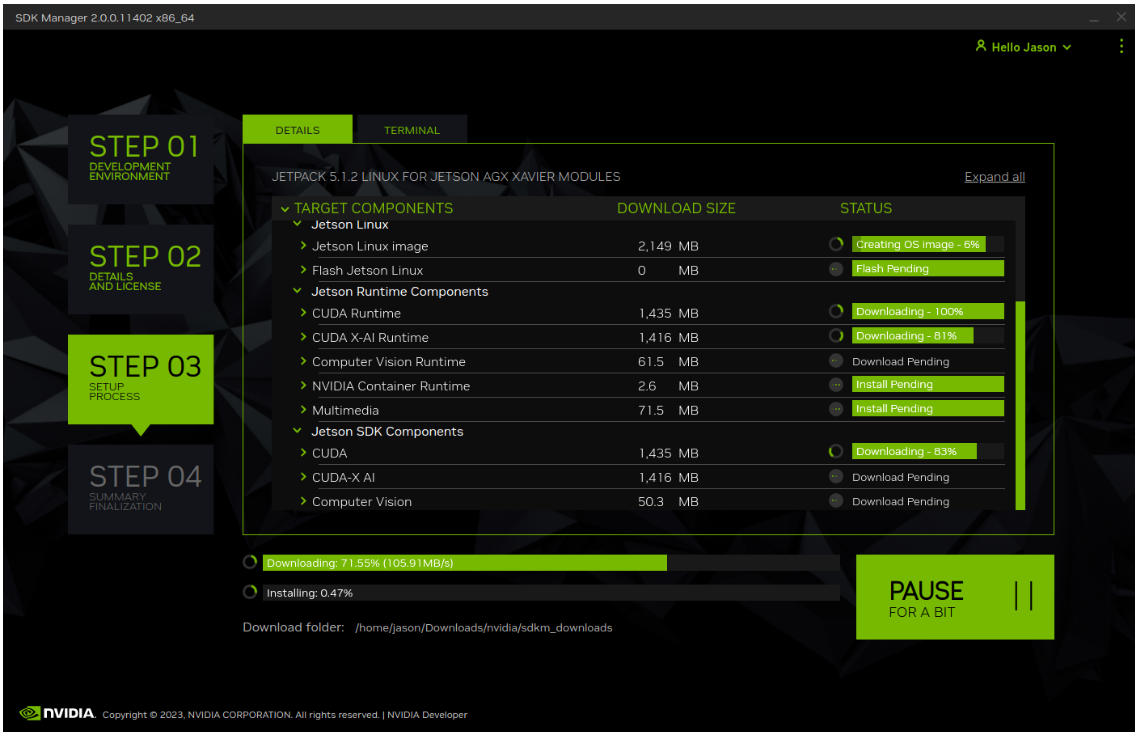
System requires up to 42GB (host) and 15GB (target) of available disk space during setup.

Download folder: /home/jason/Downloads/nvidia/sdkm_downloads change (13GB required)

Target HW image folder: /home/jason/nvidia/nvidia_sdk change (29GB required)

☑ I accept the terms and conditions of the license agreements. □ Download now. Install later.

5. Press Continue to advance to the STEP 03 – SETUP PROCESS screen and SDK Manager will begin downloading all the software components from NVIDIA, as shown below:



JETPACK 5.1.2 LINUX FOR JETSON AGX XAVIER MODULES

TARGET COMPONENTS	DOWNLOAD SIZE	STATUS
Jetson Linux		
Jetson Linux image	2,149 MB	Creating OS image - 6%
Flash Jetson Linux	0 MB	Flash Pending
Jetson Runtime Components		
CUDA Runtime	1,435 MB	Downloading - 100%
CUDA X-AI Runtime	1,416 MB	Downloading - 81%
Computer Vision Runtime	61.5 MB	Download Pending
NVIDIA Container Runtime	2.6 MB	Install Pending
Multimedia	71.5 MB	Install Pending
Jetson SDK Components		
CUDA	1,435 MB	Downloading - 83%
CUDA-X AI	1,416 MB	Download Pending
Computer Vision	50.3 MB	Download Pending

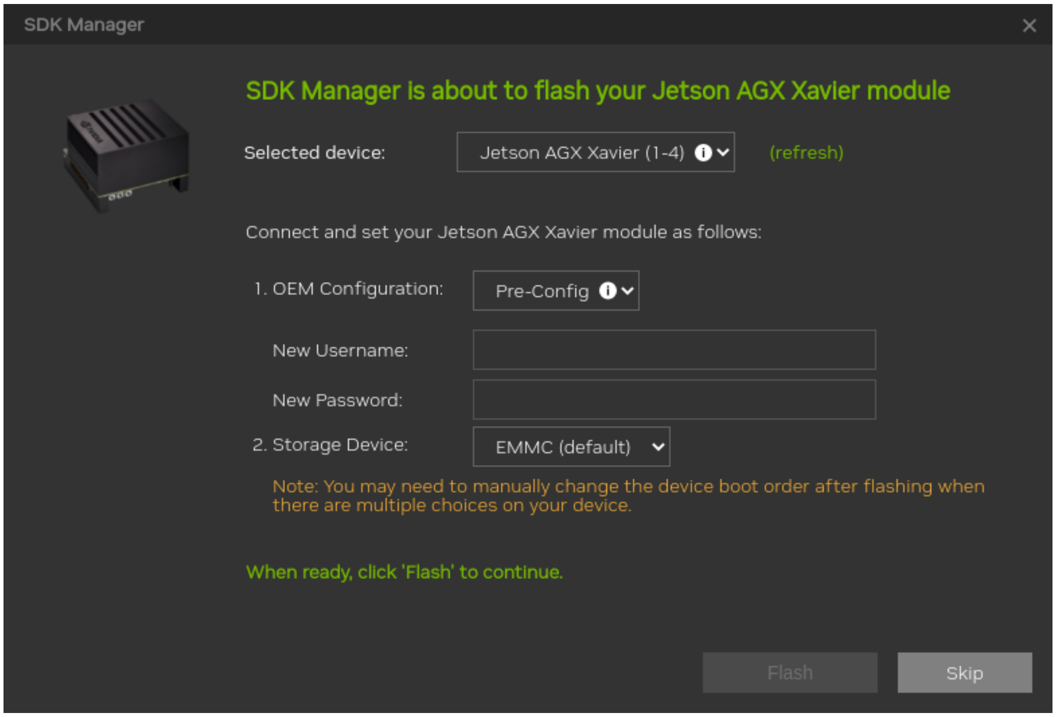
Downloading: 71.55% (105.91MB/s)

Installing: 0.47%

Download folder: /home/jason/Downloads/nvidia/sdkm_downloads

Depending on your network's speed, the download can take 30 minutes or more. During the download and setup, a detailed output log can be viewed under the TERMINAL tab.

6. Once the download and setup has completed, you will be presented with the following Flash dialog:



SDK Manager is about to flash your Jetson AGX Xavier module

Selected device: Jetson AGX Xavier (1-4) (refresh)

Connect and set your Jetson AGX Xavier module as follows:

1. OEM Configuration: Pre-Config

New Username:

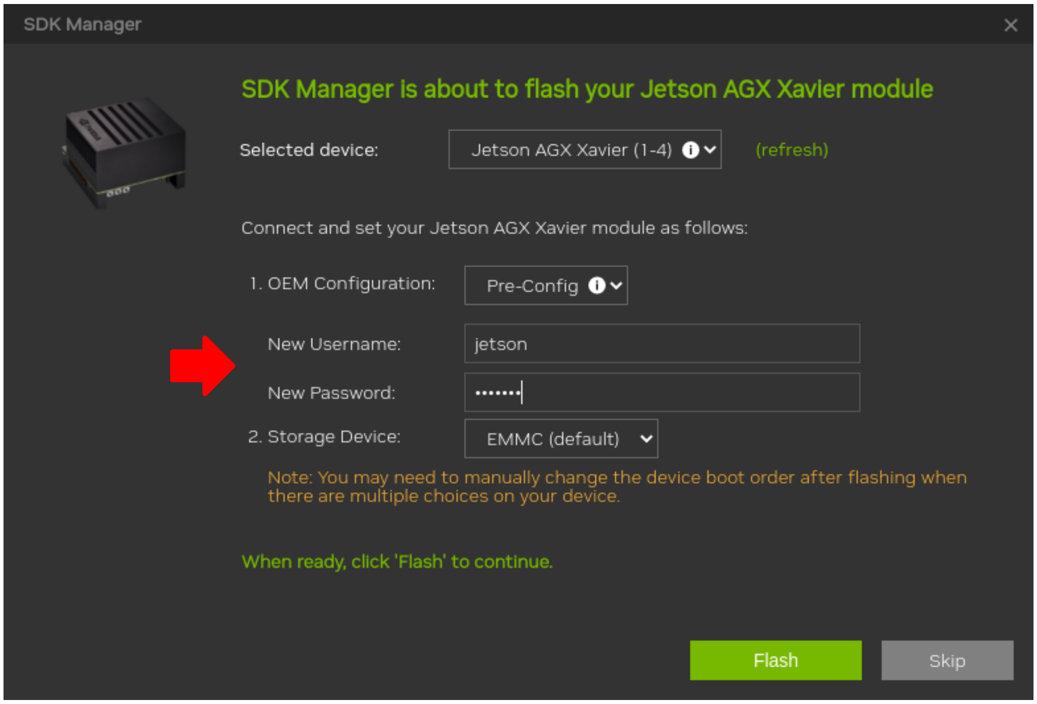
New Password:

2. Storage Device: EMMC (default)

Note: You may need to manually change the device boot order after flashing when there are multiple choices on your device.

When ready, click 'Flash' to continue.

Fill in the New Username and New Password fields for the default user that will be created on the Xavier, as shown in the following dialog:



SDK Manager

SDK Manager is about to flash your Jetson AGX Xavier module

Selected device: Jetson AGX Xavier (1-4) (refresh)

Connect and set your Jetson AGX Xavier module as follows:

1. OEM Configuration: Pre-Config

New Username: jetson

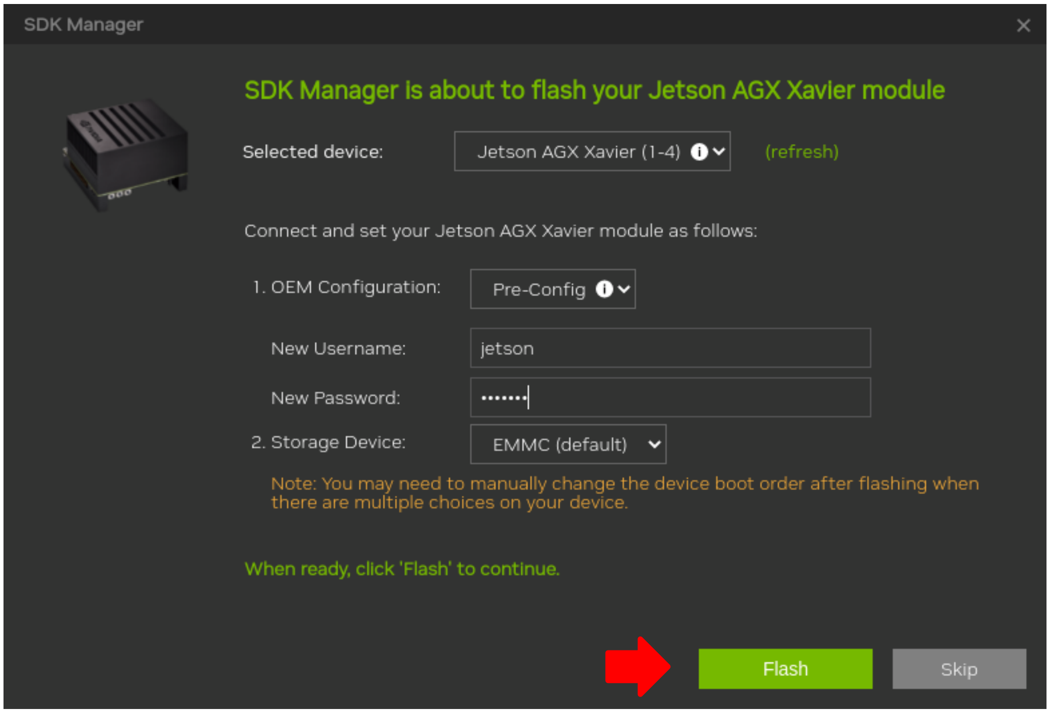
New Password: •••••••

2. Storage Device: EMMC (default)

Note: You may need to manually change the device boot order after flashing when there are multiple choices on your device.

When ready, click 'Flash' to continue.

Click the Flash button to begin flashing, as shown in the following dialog:



SDK Manager

SDK Manager is about to flash your Jetson AGX Xavier module

Selected device: Jetson AGX Xavier (1-4) (refresh)

Connect and set your Jetson AGX Xavier module as follows:

1. OEM Configuration: Pre-Config

New Username: jetson

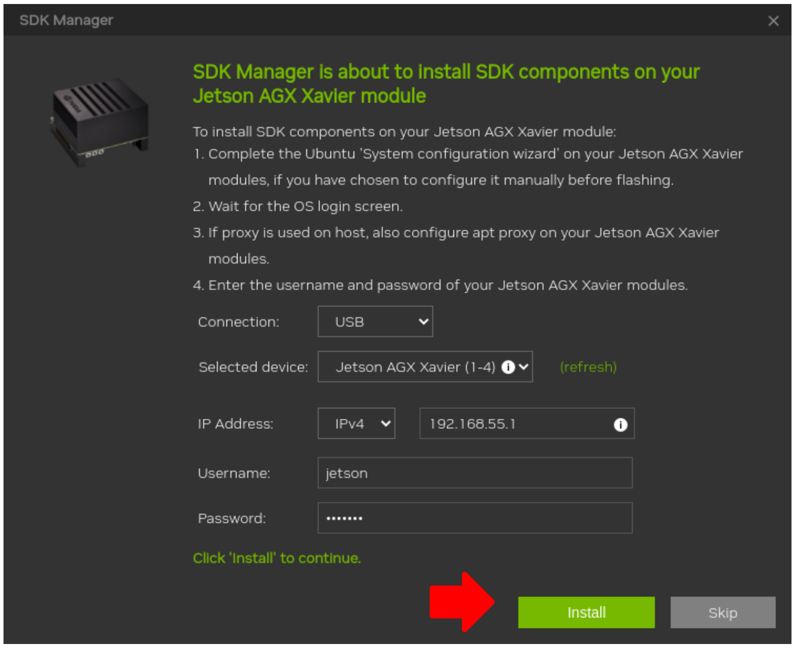
New Password: •••••••

2. Storage Device: EMMC (default)

Note: You may need to manually change the device boot order after flashing when there are multiple choices on your device.

When ready, click 'Flash' to continue.

7. During flashing the SDK Manager will prompt you to fill in Username and Password fields as shown in the following dialog:



SDK Manager

SDK Manager is about to install SDK components on your Jetson AGX Xavier module

To install SDK components on your Jetson AGX Xavier module:

1. Complete the Ubuntu 'System configuration wizard' on your Jetson AGX Xavier modules, if you have chosen to configure it manually before flashing.
2. Wait for the OS login screen.
3. If proxy is used on host, also configure apt proxy on your Jetson AGX Xavier modules.
4. Enter the username and password of your Jetson AGX Xavier modules.

Connection: USB

Selected device: Jetson AGX Xavier (1-4) (refresh)

IP Address: IPv4 192.168.55.1

Username: jetson

Password: •••••••

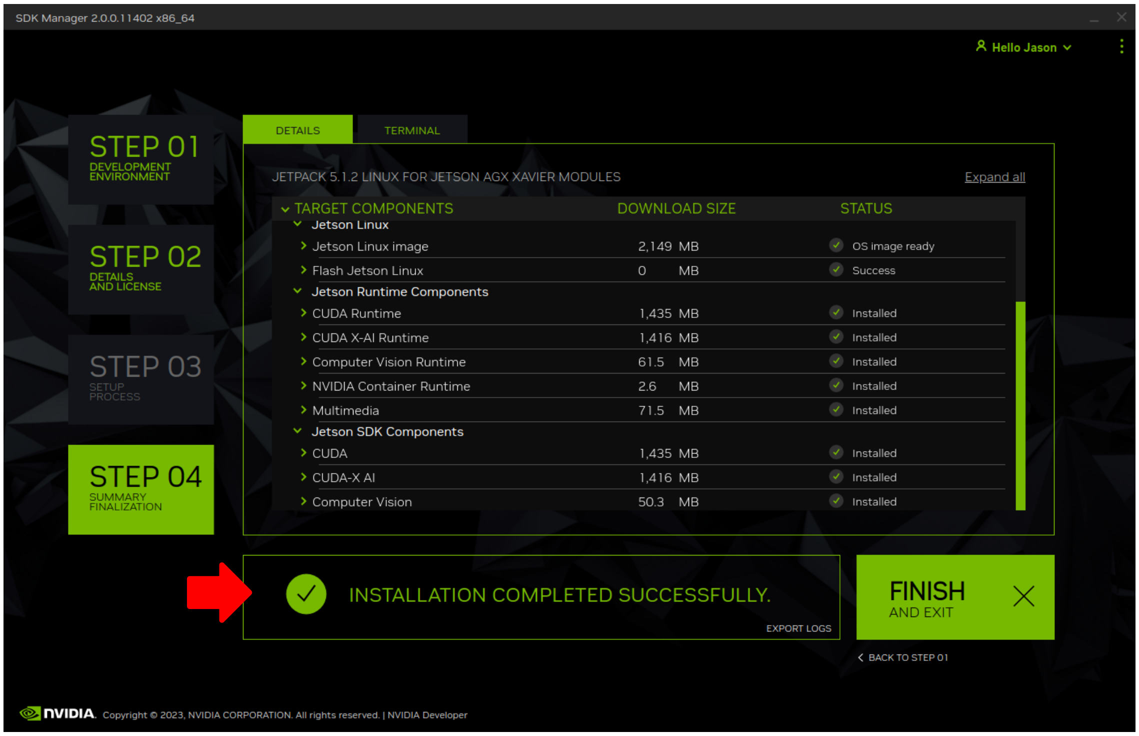
Click 'Install' to continue.

These fields default to the values specified in the previous step, so simply click Install to continue the software installation.

NOTE

To complete software installation the Xavier must be connected to the Internet.

8. Once the flashing and software installation have completed you will automatically proceed to the STEP 04 – SUMMARY FINALIZATION screen.



SDK Manager 2.0.0.11402 x86_64

JETPACK 5.1.2 LINUX FOR JETSON AGX XAVIER MODULES

TARGET COMPONENTS	DOWNLOAD SIZE	STATUS
Jetson Linux		
Jetson Linux image	2,149 MB	OS image ready
Flash Jetson Linux	0 MB	Success
Jetson Runtime Components		
CUDA Runtime	1,435 MB	Installed
CUDA X-AI Runtime	1,416 MB	Installed
Computer Vision Runtime	61.5 MB	Installed
NVIDIA Container Runtime	2.6 MB	Installed
Multimedia	71.5 MB	Installed
Jetson SDK Components		
CUDA	1,435 MB	Installed
CUDA-X AI	1,416 MB	Installed
Computer Vision	50.3 MB	Installed

INSTALLATION COMPLETED SUCCESSFULLY.

NVIDIA. Copyright © 2023, NVIDIA CORPORATION. All rights reserved. | NVIDIA Developer

Click Finish and Exit to exit the SDK Manager

3.6. Set NVIDIA Power Model to Maximum Performance

After SDK Manager has completed the Ubuntu installation on the Xavier, and while the Xavier is still booted with the NVIDIA stock kernel, log into the Xavier using the username and password chosen during flashing and run the following command:

```
$ sudo nvpmodel -m 0
```

To verify that the setting has taken effect issue the following command:

```
$ sudo nvpmodel -q
```

The following output should be displayed:

```
NV Power Mode: MAXN
0
```

This confirms that the Xavier has been correctly configured for maximum performance.

Note that once the Xavier's power mode has been set it will persist across reboots until it is explicitly changed again using the `nvpmodel` command.

3.7. Copy RedHawk packages to the Xavier

NOTE

This section assumes you are using the RedHawk 8.4.8 for Jetson AGX Xavier optical media disc. Alternatively, if you wish to install RedHawk over the network, refer to Appendix A, "Install RedHawk from Concurrent Real-Time Repositories" at the end of this document. Once installed, continue with section 3.9 below to complete installation.

After SDK Manager has completed the Ubuntu installation on the Xavier, insert the *RedHawk 8.4.8 for Jetson AGX Xavier* optical disc into the Ubuntu host's optical drive and it should automatically mount under the `/media/$USER` directory (e.g., `/media/jane/RedHawk 8.4.8 aarch64`).

Change to the mounted directory on the host and use networking to copy the RedHawk packages into the newly installed Xavier, as illustrated by the following commands:

```
$ cd /media/jane/RedHawk-8.4.8-aarch64
$ scp *.gz user@Xavier-IP-Address:/tmp
```

NOTE

Use the username, hostname and password chosen during flashing.

3.8. Install RedHawk packages on the Xavier

To install the RedHawk packages, log into the Xavier (either via ssh or via the graphical console) and issue the following commands:

```
$ cd /tmp
$ sudo tar -xf core-*.tar.gz
$ cd redhawk
$ sudo ./install
```

Enter the password for the current user if the sudo command prompts for it.

NOTE

Warnings may be displayed during package installation, including:

- *Warning: couldn't identify type of root file system for fsck hook*
- *Failed to stop turbo.service: Unit turbo.service not loaded.*
- *Failed to disable unit: Unit file turbo.service does not exist.*

You can safely ignore these messages as they are not relevant to RedHawk 8.4.8 on Jetson platform embedded installations.

At this point, all RedHawk software should be successfully installed on the Xavier, however additional steps in the next section are required before you can boot RedHawk kernels.

3.9. Add RedHawk entries to the extlinux.conf file

Perform the following steps on the Xavier to create boot entries for the RedHawk kernels:

1. Create a backup of the /boot/extlinux/extlinux.conf file.

```
$ cd /boot/extlinux
$ sudo cp extlinux.conf extlinux.bak
```

2. Open the extlinux.conf file in your preferred editor. For example, if using vi:

```
$ sudo vi extlinux.conf
```

3. Duplicate all the primary kernel entry text lines to create a second identical entry below the initial entry. For example:

```
TIMEOUT 30
DEFAULT primary

MENU TITLE L4T boot options

LABEL primary
   MENU LABEL primary kernel
   LINUX /boot/Image
   FDT /boot/dtb/kernel_tegra194-p2888-0001-p2822-0000.dtb
   INITRD /boot/initrd
   APPEND ${cbootargs} root=/dev/mmcblk0p1 ...many other boot options...

LABEL primary
   MENU LABEL primary kernel
   LINUX /boot/Image
   FDT /boot/dtb/kernel_tegra194-p2888-0001-p2822-0000.dtb
   INITRD /boot/initrd
   APPEND ${cbootargs} root=/dev/mmcblk0p1 ...many other boot options...
```

4. Customize the *first* kernel entry to point to the desired RedHawk kernel. For example, to create an entry for the RedHawk trace kernel, modify the `LABEL`, `MENU LABEL` and `LINUX` lines as follows (leaving the other lines unchanged):

```
LABEL redhawk-trace
   MENU LABEL redhawk-trace kernel
   LINUX /boot/Image-5.10.120-rt70-r35.4.1-tegra-RedHawk-8.4.8-trace
   FDT /boot/dtb/kernel_tegra194-p2888-0001-p2822-0000.dtb
   INITRD /boot/initrd
   APPEND ${cbootargs} root=/dev/mmcblk0p1 ...many other boot options...
```

NOTE

Modifying the first kernel entry will allow the NVIDIA kernel to be booted as a fallback if any typos were entered during the customization of the first entry. Failure to follow this guideline can render the Xavier unbootable and require re-flashing if a typo is accidentally introduced during editing.

5. Change the default kernel to the newly added RedHawk kernel by changing the `DEFAULT` entry at the top of the file:

```
TIMEOUT 30
DEFAULT redhawk-trace
```

NOTE

If the Xavier has been set up with a serial console, you may choose to skip this step and instead interactively decide which kernel to boot when you are presented with the boot menu on the serial console.

3.10. Verify RedHawk kernel installation

Upon reboot, the Xavier should now be running the selected RedHawk kernel. To verify this, issue the following command:

```
$ uname -r
```

You should see output displayed like the following:

```
5.10.120-rt70-r35.4.1-tegra-RedHawk-8.4.8-trace
```

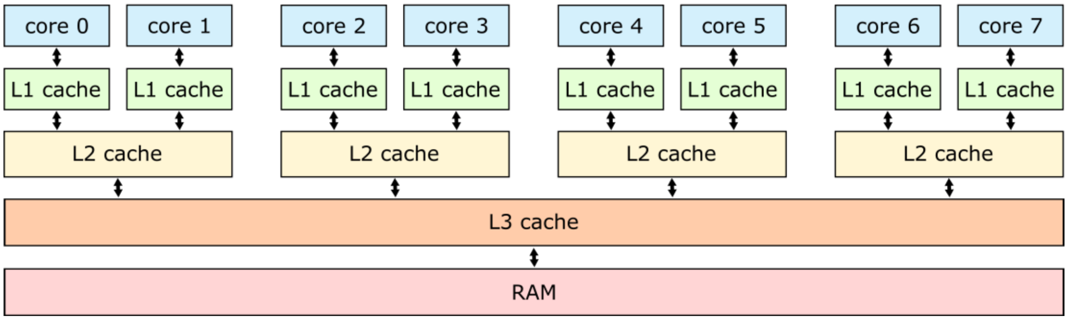
Please contact Concurrent Real-Time technical support if you had any problems during this installation (support@concurrent-rt.com or 1-800-245-6453).

4. Known Issues

Special consideration should be given to the following areas.

4.1. Xavier sibling cores can interfere with shielding

Xavier cores utilize a 3-tier cache as shown in the following diagram:



Each core has its own L1 cache, but the L2 caches are always shared between two sibling cores. In the above diagram, core 0 and core 1 are siblings, core 2 and core 3 are siblings, core 4 and core 5 are siblings, and core 6 and core 7 are siblings.

When shielding a core on the Xavier for a real-time application, activity on the shielded core's sibling core can modify the shared L2 cache and impact the determinism and performance of the real-time application. To achieve maximum real-time performance, both sibling cores must be shielded, with the unused sibling core kept idle.

4.2. Maximizing clock speeds

NVIDIA provides the `jetson_clocks` utility to control the speeds of various hardware clocks on the Xavier. This utility provides several options and must be run as the root user. Invoke the utility with the `--show` option to view all current clock speeds. For example:

```
jetson@ubuntu:~$ sudo jetson_clocks --show 2>/dev/null
SOC family:tegra194  Machine:Jetson-AGX
Online CPUs: 0-7
cpu0:  Online=1 Governor=schedutil MinFreq=1190400 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=1 c6=1
cpu1:  Online=1 Governor=schedutil MinFreq=1190400 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=1 c6=1
cpu2:  Online=1 Governor=schedutil MinFreq=1190400 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=1 c6=1
cpu3:  Online=1 Governor=schedutil MinFreq=1190400 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=1 c6=1
cpu4:  Online=1 Governor=schedutil MinFreq=1190400 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=1 c6=1
cpu5:  Online=1 Governor=schedutil MinFreq=1190400 MaxFreq=2265600 CurrentFreq=2112000 IdleStates: C1=1 c6=1
cpu6:  Online=1 Governor=schedutil MinFreq=1190400 MaxFreq=2265600 CurrentFreq=2035200 IdleStates: C1=1 c6=1
cpu7:  Online=1 Governor=schedutil MinFreq=1190400 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=1 c6=1
GPU MinFreq=114750000 MaxFreq=1377000000 CurrentFreq=114750000
EMC MinFreq=204000000 MaxFreq=2133000000 CurrentFreq=2133000000 FreqOverride=0
DLA0_CORE:   Online=1 MinFreq=0 MaxFreq=1395200000 CurrentFreq=1395200000
DLA0_FALCON: Online=1 MinFreq=0 MaxFreq=844800000 CurrentFreq=844800000
DLA1_CORE:   Online=1 MinFreq=0 MaxFreq=1395200000 CurrentFreq=1395200000
DLA1_FALCON: Online=1 MinFreq=0 MaxFreq=844800000 CurrentFreq=844800000
PVA0_VPS0: Online=1 MinFreq=0 MaxFreq=1088000000 CurrentFreq=1088000000
PVA0_VPS1: Online=1 MinFreq=0 MaxFreq=1088000000 CurrentFreq=1088000000
PVA0_AXI:  Online=1 MinFreq=0 MaxFreq=844800000 CurrentFreq=844800000
PVA1_VPS0: Online=1 MinFreq=0 MaxFreq=1088000000 CurrentFreq=1088000000
PVA1_VPS1: Online=1 MinFreq=0 MaxFreq=1088000000 CurrentFreq=1088000000
PVA1_AXI:  Online=1 MinFreq=0 MaxFreq=844800000 CurrentFreq=844800000
CVNAS MinFreq=0 MaxFreq=1356800000 CurrentFreq=1356800000
FAN Dynamic Speed control=active hwmon3_pwm1=0
NV Power Mode: MAXN
jetson@ubuntu:~$
```

Notice that, by default, some of the current frequency values above are significantly lower than the maximum frequencies supported by the hardware.

To achieve the best performance, set all clock speeds to their maximum values by invoking the `jetson_clocks` utility without any options. Then, to verify that all clock speeds have been maximized, run the `jetson_clocks` utility again with the `--show` option. For example:

```
root@ubuntu:~# jetson_clocks
root@ubuntu:~# jetson_clocks --show 2>/dev/null
SOC family:tegra194  Machine:Jetson-AGX
Online CPUs: 0-7
cpu0:  Online=1 Governor=schedutil MinFreq=2265600 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=0 c6=0
cpu1:  Online=1 Governor=schedutil MinFreq=2265600 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=0 c6=0
cpu2:  Online=1 Governor=schedutil MinFreq=2265600 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=0 c6=0
cpu3:  Online=1 Governor=schedutil MinFreq=2265600 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=0 c6=0
cpu4:  Online=1 Governor=schedutil MinFreq=2265600 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=0 c6=0
cpu5:  Online=1 Governor=schedutil MinFreq=2265600 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=0 c6=0
cpu6:  Online=1 Governor=schedutil MinFreq=2265600 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=0 c6=0
cpu7:  Online=1 Governor=schedutil MinFreq=2265600 MaxFreq=2265600 CurrentFreq=2265600 IdleStates: C1=0 c6=0
GPU MinFreq=1377000000 MaxFreq=1377000000 CurrentFreq=1377000000
EMC MinFreq=204000000 MaxFreq=2133000000 CurrentFreq=2133000000 FreqOverride=1
DLA0_CORE:   Online=1 MinFreq=0 MaxFreq=1395200000 CurrentFreq=1395200000
DLA0_FALCON: Online=1 MinFreq=0 MaxFreq=844800000 CurrentFreq=844800000
DLA1_CORE:   Online=1 MinFreq=0 MaxFreq=1395200000 CurrentFreq=1395200000
DLA1_FALCON: Online=1 MinFreq=0 MaxFreq=844800000 CurrentFreq=844800000
PVA0_VPS0: Online=1 MinFreq=0 MaxFreq=1088000000 CurrentFreq=1088000000
PVA0_VPS1: Online=1 MinFreq=0 MaxFreq=1088000000 CurrentFreq=1088000000
PVA0_AXI:  Online=1 MinFreq=0 MaxFreq=844800000 CurrentFreq=844800000
PVA1_VPS0: Online=1 MinFreq=0 MaxFreq=1088000000 CurrentFreq=1088000000
PVA1_VPS1: Online=1 MinFreq=0 MaxFreq=1088000000 CurrentFreq=1088000000
PVA1_AXI:  Online=1 MinFreq=0 MaxFreq=844800000 CurrentFreq=844800000
CVNAS MinFreq=0 MaxFreq=1356800000 CurrentFreq=1356800000
FAN Dynamic Speed control=active hwmon3_pwm1=0
NV Power Mode: MAXN
root@ubuntu:~#
```

Notice that all the current frequency values above have been configured to their maximum speeds.

NOTE

Unlike `nvpmodel`, clock speed changes made by `jetson_clocks` do <u>not</u> persist across a reboot. Thus, you must always run `jetson_clocks` again after a reboot to ensure that all clock speeds are maximized.

4.3. Xavier hardware not optimized for real-time at boot

By default, at boot time the Xavier hardware is optimized for maximum throughput instead of maximum real-time performance; in this mode real-time applications will experience less determinism and higher latencies during execution.

To run real-time workloads on the Xavier with high determinism and significantly lower latencies, the `rt_safe_mask` control file gives users the ability to select the desired level of real-time performance on the Xavier. For example, invoking the following command as the root user will maximize real-time performance:

```
# echo 255 > /sys/kernel/debug/tegra_mce/rt_safe_mask
```

The above command enables maximum real-time performance on all 8 CPU cores on the Xavier, and this is required to achieve maximum real-time performance of the whole system. Note that 255 is a bit mask (i.e., hex 0xFF or binary 0b11111111) indicating that each of the 8 individual CPU cores is to be configured for maximum real-time performance.

Users may choose to exclude one or more CPU cores from the bit mask to sacrifice a degree of real-time performance in exchange for improved throughput; for example, invoking the following command as the root user excludes CPU core 0 from the bit mask:

```
# echo 254 > /sys/kernel/debug/tegra_mce/rt_safe_mask
```

The above command will still improve real-time performance compared to the default setting of `0` (where no CPU cores are optimized for real-time) but the improvement will not be as good as when all CPU cores are enabled for real-time. However, in this mode throughput will improve.

Finally, users can also return to the default maximum throughput state of the Xavier (where no CPU cores are optimized for real-time) by invoking the following command as the root user:

```
echo 0 > /sys/kernel/debug/tegra_mce/rt_safe_mask
```

NOTE

Because the Xavier defaults to optimizing for maximum throughput, it may be advantageous to start real-time applications and let them run for some amount of time before setting `rt_safe_mask` to `255` to maximize real-time performance; users are encouraged to experiment with this technique to determine whether any advantages may be gained.

4.4. Maximizing fan speed during heavy loads

By default, the speed of Xavier's CPU cooling fan will be dynamically adjusted, as needed, to always ensure proper cooling. This default dynamic fan behavior should be more than adequate for almost all use cases.

However, for periods of prolonged benchmarking and/or stress testing, Concurrent recommends maximizing the CPU cooling fan speed before starting tests to prevent fast temperature rises. Temperatures that rise too quicky may result in slowing down CPU speeds, which can negatively impact benchmarking.

Invoke `jetson_clocks` with the `--fan` option to maximize the speed of the CPU cooling fan.

4.5. Entropy gathering daemon interferes with real-time

Ubuntu 20.04 enables the Hardware Volatile Entropy Gathering daemon (`haveged`) that continuously samples various hardware resources (e.g., cache, branch predictions, memory translation tables) to collect entropy and improve random number generation.

This daemon's constant hardware probing activity can negatively impact real-time performance even on shielded CPU cores, and the daemon should be stopped before starting real-time workloads by invoking the following command as the root user:

```
# systemctl stop haveged.service
```

Do not disable the entropy gathering service because various Ubuntu startup services require a minimum level of entropy to be gathered before they will start; this includes services required for the X Server to properly initialize. However, after the Xavier has started with graphics displaying you can safely stop the service to improve real-time performance.

4.6. Debug kernel not supplied with RedHawk 8.4.8

The RedHawk debug kernel is not provided with RedHawk 8.4.8 due to NVIDIA incompatibilities with various debug-related kernel configuration options. Concurrent is working with NVIDIA to address this issue and hopefully it will be corrected in a future RedHawk release.

4.7. RedHawk kernels and OEM verification failures

Booting any of the RedHawk kernels on the Xavier will result in the following warnings being displayed almost immediately after the boot process begins:

```
[0014.855] I> Validate kernel ...
[0014.855] I> T19x: Authenticate kernel (bin_type: 37), max size 0x5000000
[0014.856] E> Stage2Signature validation failed with SHA2!!
[0014.856] C> OEM authentication of kernel header failed!
[0014.857] W> Failed to validate kernel binary (err=1077936152, fail=1)
[0014.857] W> Security fuse not burned, continue to boot
```

These warnings can be safely ignored because OEM security fuses are not burned by default on Jetson hardware platforms.

For more information, refer to the *NVIDIA Jetson AGX Xavier Fuse Specification* document that is available on the NVIDIA Developer website.

4.8. Change to PAM Capability module directory

The PAM Capability module directory that is documented in some versions of the *RedHawk User's Guide* is incorrect for the Xavier. The correct PAM Capability module location should be shown as:

```
/lib/aarch64-linux-gnu/security/pam_capability.so
```

If you wish to use the PAM Capability feature with services in the `/etc/pam.d` directory, then you should refer to the PAM Capability module as follows:

```
session required /lib/aarch64-linux-gnu/security/pam_capability.so
```

4.9. Custom kernels will not install kernel Image files

When custom kernels are built and installed on the Xavier, the `make install` step will install a custom `vmlinuz-…` file into `/boot`, but it will *not* install a matching custom `Image-…` file.

The newly installed custom `vmlinuz-…` file path can simply be entered for the custom kernel entry's `LINUX` line in `extlinux.conf` to reference the newly installed custom kernel. Alternatively, the newly built uncompressed `Image-…` file can be manually copied to `/boot` with the following command:

```
sudo cp ./arch/arm64/boot/Image /boot/Image-$(make ccur-kernel-release)
```

The custom kernel entry's `LINUX` line can then reference the copied custom `Image-…` file.

A. Install RedHawk from Concurrent Real-Time Repositories

RedHawk can be installed over the network utilizing the Concurrent Real-Time software repositories. This has the advantage that it will install the latest updates that have been released, however some up-front setup is required to perform the installation; follow the steps below to complete this setup.

A.1. Install Concurrent Real-Time Public Keys

Concurrent Real-Time signs its APT repositories with a GPG-generated DSA and ElGamal key pair. You need to install Concurrent Real-Time's public key so that the APT system can authorize usage of the repositories.

You can download the ccur-public-keys file and import the key file as needed into APT's key ring. You should import the key file before attempting to access software from Concurrent Real-Time's repositories.

The following commands will download the public key and install it.

```
wget -q http://redhawk.concurrent-rt.com/network/ccur-public-keys
apt-key add ccur-public-keys
```

If you wish to check the authenticity of the public keys file you downloaded from Concurrent Real-Time's web site, request the key fingerprints from Concurrent Software Support and compare them to the fingerprints of the keys you downloaded.

You can obtain the fingerprints of the keys you downloaded with the following command:

```
gpg --with-fingerprint ccur-public-keys
```

A.2. Install Repository Definition Files

Repository definition files tell the APT installation and update system how to access specific repositories. The definition files must reside in the `/etc/apt/sources.list.d/` directory and the file name must end `.list`; e.g. `/etc/apt/sources.list.d/ccur.list`.

For example, use the following format for the repository definition of the RedHawk 8.4.8 for Jetson AGX Xavier product:

```
deb [arch=arm64] http://redhawk.concurrent-rt.com/ubuntu/login/passwd/redhawk/agx-xavier 8.4.8 core
```

For example, the following entry substitutes a fictional login and password:

```
deb [arch=arm64] http://redhawk.concurrent-rt.com/ubuntu/L12345/xyzzy/redhawk/agx-xavier 8.4.8 core
```

Note that if the NightStar tools have been purchased for this machine, then you should also add a repository definition for the NightStar tools below the previously added line. For example:

```
deb [arch=arm64] http://redhawk.concurrent-rt.com/ubuntu/L12345/xyzzy/nightstar 5.1 rt ubu20
```

Once the repository definition file is in place the repository setup is complete. Run the following command as root to ensure that the software repositories have been added correctly:

```
sudo apt update
```

If you see errors or warnings produced that reference the newly added `ccur.list` file, then verify that all the above steps have all been completed as documented.

A.3. Install RedHawk Packages

Now that the setup is done, simply issue the following command to install all the latest RedHawk 8.4.8 packages (and NightStar 4.6 packages if purchased) onto the current system:

```
apt install 'ccur-*'
```

Once this command completes the RedHawk packages are installed. Continue with Section 3.8, “Add RedHawk entries to the `extlinux.conf` file," to finalize the RedHawk installation.

A.4. Support

If you need assistance, please contact the Concurrent Real-Time Software Support Center at our toll-free number 1-800-245-6453. For calls outside the continental United States, the number is 1-954-283-1822. The Software Support Center operates Monday through Friday from 8 a.m. to 5 p.m., Eastern Standard Time.

You may also submit a request for assistance at any time by using the Concurrent Real-Time website at http://concurrent-rt.com/support or by sending an email to support@concurrent-rt.com.