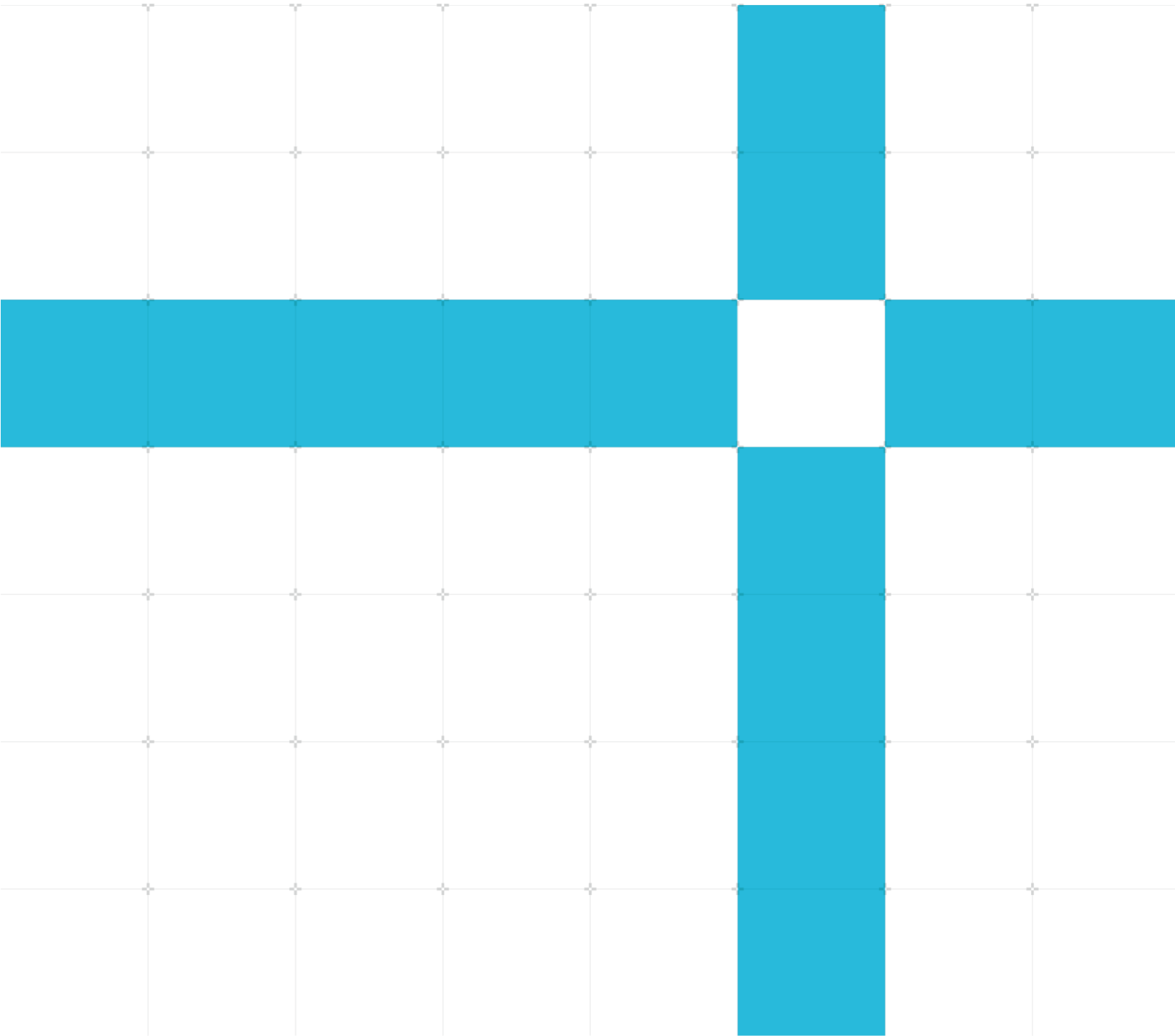
Arm® Mobile Studio 2023.2

Product revision: r23p2-00rel0

Release Note

Non-Confidential

Issue 00
DSHVE-DC-06002



Arm Mobile Studio 2023.2

Release Note

Non-Confidential Proprietary Notice

Arm Limited. Company 02557590 registered in England.
110 Fulbourn Road, Cambridge, England CB1 9NJ.
(LES-PRE-20349)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm Mobile Studio, create a ticket on https://support.developer.arm.com.

To provide feedback on the document, fill the following survey:
https://developer.arm.com/documentation-feedback-survey.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

To report offensive language in this document, email terms@arm.com.

Contents

1 Release overview

The following sections describe the product and its quality status at time of release.

1.1 Product description

Arm® Mobile Studio is a tool suite enabling Android application developers to detect performance bottlenecks in their Arm CPU software and Arm Immortalis™ and Arm Mali™ GPU rendering. Profiling is provided through analysis of performance counters from the hardware, and the target application's graphics API usage.

This release of Arm Mobile Studio includes:

- Streamline, for profiling application software and rendering performance. Streamline now integrates Performance Advisor, a reporting tool used for automating rendering performance analysis and reporting in continuous integration deployments.
- Graphics Analyzer, for debugging and inspecting usage of graphics APIs.
- Mali Offline Compiler, for static analysis of shader programs and compute kernels.

1.1.1 Component versions

This release of Arm Mobile Studio includes the following tool versions:

- Streamline 8.6
- Graphics Analyzer 5.12.1
- Mali Offline Compiler 8.0

1.2 Release status

This is the REL quality release of the Arm Mobile Studio 2023.2 (r23p2-00rel0) software.

1.3 Changes in this release

This release of Arm Mobile Studio contains the following changes.

1.3.1 Streamline

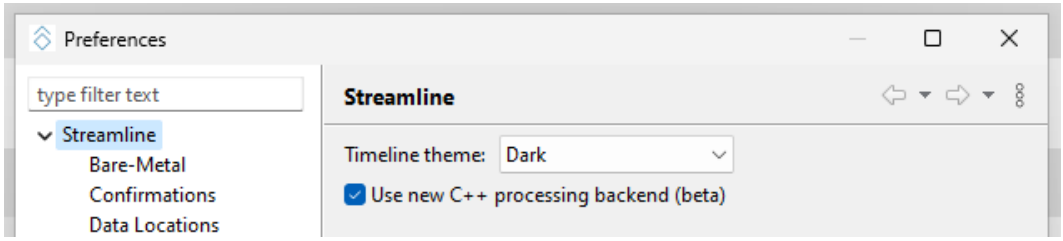
Streamline has the following changes:

- Streamline now supports the Arm Immortalis-G720, Mali-G720, and Mali-G620 GPUs.

- Streamline counter templates for the Arm Midgard GPUs have been refreshed. The new templates use the same design and naming conventions as the later Arm GPU architecture generations and are supported by matching performance counter guides.
- The Streamline CI helper scripts, `gator_me.py` and `lwi_me.py`, have been merged into a combined `streamline_me.py` script.
    - This script uses the same command-line arguments as the previous `lwi_me.py` script.
    - The default value of `--lwi-mode` has changed to `off`. Set this argument to `counters` to restore the previous behavior.
- Performance Advisor analysis will now automatically clip to the end of the capture if `--clip-end` exceeds the length of the capture, instead of reporting an error.
- The Performance Advisor light-weight interceptor now supports counting dynamic render passes created using `vkCmdBeginRendering[KHR]()`.
- Fix: Streamline GPU counter sampling now includes a workaround for physical memory fragmentation on older Mali devices, although may still fail to allocate a sample buffer if physical memory fragmentation is severe. If data capture fails, reboot your device and try again.
- Fix: The Mali timeline has improved queue naming for Arm GPUs with a command stream frontend.
- Fix: The Mali timeline correctly handles visualization for multi-process applications that fork a child process during rendering.
- Fix: The Performance Advisor light-weight interceptor now correctly counts render passes and draw calls in command buffers using implicit reset on `vkBeginCommandBuffer()`.

Streamline 8.4 introduced a new data analysis backend which significantly improves capture analysis performance and reduces memory footprint. This new backend is enabled by default but can be turned off if problems are encountered.

- Via the GUI, the C++ analysis backend can be disabled by unticking the "Use new C++ processing backend (beta)" tick box in the Preferences.



- Via the command line, the C++ analysis backend can be disabled by setting the `STREAMLINE_ENABLE_DBV3_DEMO` environment variable to the value `false`.

  If you encounter any problems with the C++ backend, please contact the product team via the mobilestudio@arm.com email address, so that we can investigate and fix the issue for a future release.

1.3.2 Graphics Analyzer

Graphics Analyzer has the following changes:

- No changes in this release.

1.3.3 Mali Offline Compiler

Mali Offline Compiler has the following changes:

- Vertex shader attributes in the stream recommendation report are now sorted to improve readability. Attributes with explicit locations are shown first, in location order, and then any remaining attributes with implicit locations are shown in string sort order.
- The bundled `glslangValidator` tool, used for compiling Vulkan shaders from GLSL source, has been updated to the latest version published by Khronos.
- Fix: The incorrect compiler warning “Warning: 'Binding' is larger than allowed maximum" is no longer generated for Vulkan shaders.
- Fix: Temporary directories that are created when compiling shaders are now removed when the tool exits.

1.4 Known issues in this release

This release of Arm Mobile Studio contains the following known issues.

1.4.1 Streamline

Streamline has the following known issues:

- SDDAP-12290: The Mali DDK can fail to emit the Perfetto data required for the scheduling timeline visualization. This results in periods that show as idle in the scheduler timeline when the GPU is clearly active in the counter data. This is fixed in the latest Mali r44p1 driver.
- SDDAP-12293: The Mali GPU counter visualization templates used in the Streamline timeline are not pattern matching Vulkan software counters generated by the Performance Advisor layer driver, so only OpenGL ES software counters are shown by default.
- SDDAP-12294: The Performance Advisor layer driver is not counting Vulkan render passes created with `vkCmdBeginRenderPass2[KHR]()` in the render pass software counter.
- SDDAP-11607: macOS host installs do not show Arm disassembly views.
- SDDAP-11426: Linux host installs using NVIDIA drivers can experience areas of the UI rendering as black rectangles when using monitor scaling. This can be worked around by setting the environment variable GDK_SCALE to 1 before launching Streamline. For ease of use, this can be set in the Streamline launch script.

2 Support

To help you get started we provide a number of quick start guides available online:

- Getting Started Guides on developer.arm.com

Technical support for Arm Mobile Studio is provided via our developer forums:

- Developer forums on community.arm.com

2.1 How-to videos

Refer to the following videos to learn how to use Arm Mobile Studio tools.

- Streamline
- Performance Advisor
- Graphics Analyzer
- Mali Offline Compiler

To learn more about Mali GPUs and how to develop optimized graphics content for mobile devices, refer to the **Mali GPU Training Series**.

2.2 Host OS support

This release has been developed for the following host operating systems:

Table 2-1: Host operating system used in developing this release

Operating system	Version
Windows	10 or newer
macOS	10.15 (Catalina) or newer
Ubuntu Linux	18.04 (Bionic Beaver) or newer

2.3 Target OS support

This release has been developed for the following target operating systems:

Table 2-2: Target operating system used in developing this release

Feature	Version
Streamline	Android 8 or newer
Streamline Performance Advisor for OpenGL ES applications	Android 8 or newer with manual annotation Android 10 or newer to use the Light-weight Interceptor

Streamline Performance Advisor for Vulkan applications	Android 9 or newer
Graphics Analyzer OpenGL ES	Android 8 or newer
Graphics Analyzer Vulkan	Android 9 or newer

2.4 Related projects

Arm provides several open-source projects that can be used by application developers as part of their application development.

2.4.1 Mobile Studio for Unity package

Current version: 1.5.0 (September 2022)

The Mobile Studio for Unity package provides an open-source Unity game engine integration for Streamline and Performance Advisor. The package provides:

- C# bindings for Streamline's annotation API, allowing users to export custom software counters, and event annotations.
- Integration with the Unity profiler data source, exporting Unity object counts and memory allocations as custom software counters.

The annotation API provides a generic means to markup a Streamline capture. It can be used to emit the semantic tags that Performance Advisor reports use to denote interesting gameplay regions.

Recent changes:

- None.

The package is available on GitHub and can be imported directly into your Unity project using the Unity package manager. See the GitHub project documentation for more details.

- https://github.com/ARM-software/mobile-studio-integration-for-unity/

2.4.2 ASTC Encoder texture compressor

Current version: 4.4.0 (March 2023)

The Arm ASTC Encoder (astcenc) is an open-source texture compressor for the Adaptive Scalable Texture Compression (ASTC) texture format. It supports all block sizes, all color profiles, as well as both 2D and volumetric 3D textures.

The astcenc compressor can be built as either a standalone command line application or a library that can be integrated into an existing asset creation pipeline.

Recent changes:

- None.

The source code is available on GitHub, in addition to binary releases of the command line utility for Windows, macOS, and Linux.

- https://github.com/ARM-software/astc-encoder

2.4.3 HWCPipe library

Current version: 2.0.0 (June 2023)

The Hardware Counter Pipe (HWCPipe) library is an open-source utility that allows applications to select and sample a set of Arm GPU performance counters. This library provides access to the same counter data that can be visualized in the Streamline tool, allowing integration of Arm GPU data into custom tooling.

The current release of the library only supports sampling of physical hardware counters. The v2.1 update, which is due later in the summer, will allow applications to request samples of derived counters. For derived counters the library will automatically sample from the required hardware counters and perform the necessary computation to return the derived metric.

Recent changes:

- The recently released HWCPipe 2.0 is a major update of the library, providing a significant update to the capabilities of the tool. This version supports all Arm GPUs from the Mali-T700 series onwards, and exposes the full set of hardware performance counters, providing GPU data parity with the Streamline tool. Sampling CPU counters is no longer supported.

The source code is available on GitHub:

- https://github.com/ARM-software/HWCPipe

2.4.4 libGPUInfo library

Current version: 1.0.0 preview (May 2023)

The libGPUInfo library is an open-source utility that can be integrated into an application to query the configuration of the Arm GPU present in the system, including the GPU model, shader core count, shader core performance characteristics, and cache size. This information can be used to adjust the application workload at runtime to match the capabilities of the device being used.

Recent changes:

- libGPUInfo now supports the Arm Immortalis-G720, Mali-G720, and Mali-G620 GPUs.

The source code is available on GitHub:

- https://github.com/ARM-software/libGPUInfo

3 Installation

This section describes how to install and configure Arm Mobile Studio to run on 64-bit Windows, macOS®, and Linux.

Mobile Studio requires Android Debug Bridge (ADB) and Python 3.5 (or newer), to enable connection to your device. Make sure you have these tools installed and that you have configured your environment to use them.

3.1 Install on Windows

Arm Mobile Studio is provided with an installer executable. Double-click the .exe file and follow the instructions in the setup wizard.

- To launch Streamline, open the Windows Start menu, navigate to the Arm Mobile Studio folder, and select the "Arm MS Streamline 2023.2" shortcut,
- To launch Graphics Analyzer, open the Windows Start menu, navigate to the Arm Mobile Studio folder, and select the "Arm MS Graphics Analyzer 2023.2" shortcut.
- Performance Advisor is a command-line tool that is part of the Streamline application. To use it to generate a performance report, you must first run the provided Python script to enable Streamline to collect frame data from the device. This process is described in detail in the Get started with Performance Advisor tutorial.

  Once you have captured a profile with Streamline, run the `Streamline-cli -pa` command on the Streamline capture file. This command is added to your PATH environment variable during installation, so it can be used from anywhere.

  ```
  Streamline-cli.exe -pa <options> my_capture.apc
  ```

- To launch Mali Offline Compiler, open a command terminal, navigate to your work directory, and run the `malioc` command on a shader program. The `malioc` command is added to your PATH environment variable during installation, so can be used from anywhere.

  ```
  malioc.exe <options> my_shader.frag
  ```

3.2 Install on macOS

Arm Mobile Studio is provided as a .dmg package. To mount it, double-click the .dmg package and follow the instructions. The Mobile Studio directory tree is copied to the Applications directory on your local file system for easy access.

Launch the tools directly from the Arm Mobile Studio directory tree in your Applications directory.

- To launch Streamline, go to the <installation_directory>/streamline directory, and open the Streamline.app file.
- To launch Graphics Analyzer, go to the <installation_directory>/graphics_analyzer/gui directory and open the Graphics Analyzer.app file.

- To launch Performance Advisor, go to the **<installation_directory>/streamline** directory, and double-click the **Streamline-cli-launcher** file. Your computer will ask you to allow Streamline to control the Terminal application. Allow this.

  The Performance Advisor launcher opens the Terminal application and updates your PATH environment variable so you can run Performance Advisor from any directory.

  Performance Advisor is a command-line tool that is part of the Streamline application. To use it to generate a performance report, you must first run the provided Python script to enable Streamline to collect frame data from the device. This process is described in detail in the Get started with Performance Advisor tutorial.

  Once you have captured a profile with Streamline, run the **`Streamline-cli -pa`** command on the Streamline capture file to generate a performance report:

  ```
  Streamline-cli -pa <options> my_capture.apc
  ```

- To launch Mali Offline Compiler, go to the **<installation_directory>/mali_offline_compiler** directory, and double-click the **mali_offline_compiler_launcher** file.

  The Mali Offline Compiler launcher opens the Terminal application and updates your PATH environment variable so you can run the **`malioc`** command from any directory.

  To generate a shader analysis report, run the **`malioc`** command on a shader program:

  ```
  malioc <options> my_shader.frag
  ```

On some versions of macOS, you might see a message that Mali Offline Compiler is not recognized as an application from an identified developer. To enable Mali Offline Compiler, cancel this message, then open **System Preferences > Security and Privacy**, and select **Allow Anyway** for the **`malioc`** application.

3.3 Install on Linux

Arm Mobile Studio is provided as a gzipped tar archive. Extract this tar archive to your preferred location, using a recent version (1.13 or later) of GNU tar:

```
tar xvzf Arm_Mobile_Studio_2023.2_linux.tgz
```

Launch the tools directly from the location where you extracted the package.

- To launch Streamline, go to the **<installation_directory>/streamline** directory and run the **`Streamline`** file.

  ```
  cd <installation_directory>/streamline
  ./Streamline
  ```

- To launch Graphics Analyzer, go to the **<installation_directory>/graphics_analyzer/gui** directory and run the **aga** file.

  ```
  cd <installation_directory>/graphics_analyzer/gui
  ./aga
  ```

- Performance Advisor is a command-line tool that is part of the Streamline application. To use it to generate a performance report, you must first run the provided Python script to enable Streamline to collect frame data from the device. This process is described in detail

in the Get started with Performance Advisor tutorial.

Once you have captured a profile with Streamline, go to the <installation_directory>/streamline directory and run the `Streamline-cli -pa` command on the Streamline capture file to generate a performance report:

```
cd <installation_directory>/performance_advisor
./Streamline-cli -pa <options> my_capture.apc
```

- To launch Mali Offline Compiler, go to the <installation_directory>/mali_offline_compiler directory and run the `malioc` command on a shader program.

```
cd <installation_directory>/mali_offline_compiler
./malioc <options> my_shader.frag
```

You might find it useful to edit your PATH environment variable to add the paths to the `Streamline-cli` and `malioc` executables so that you can run them from any directory. Add the following commands to the .bashrc file in your home directory, so that they are set whenever you initialize a shell session:

```
PATH=$PATH:/<installation_directory>/streamline
PATH=$PATH:/<installation_directory>/mali_offline_compiler
```