



Arm® Mobile Studio 2023.4

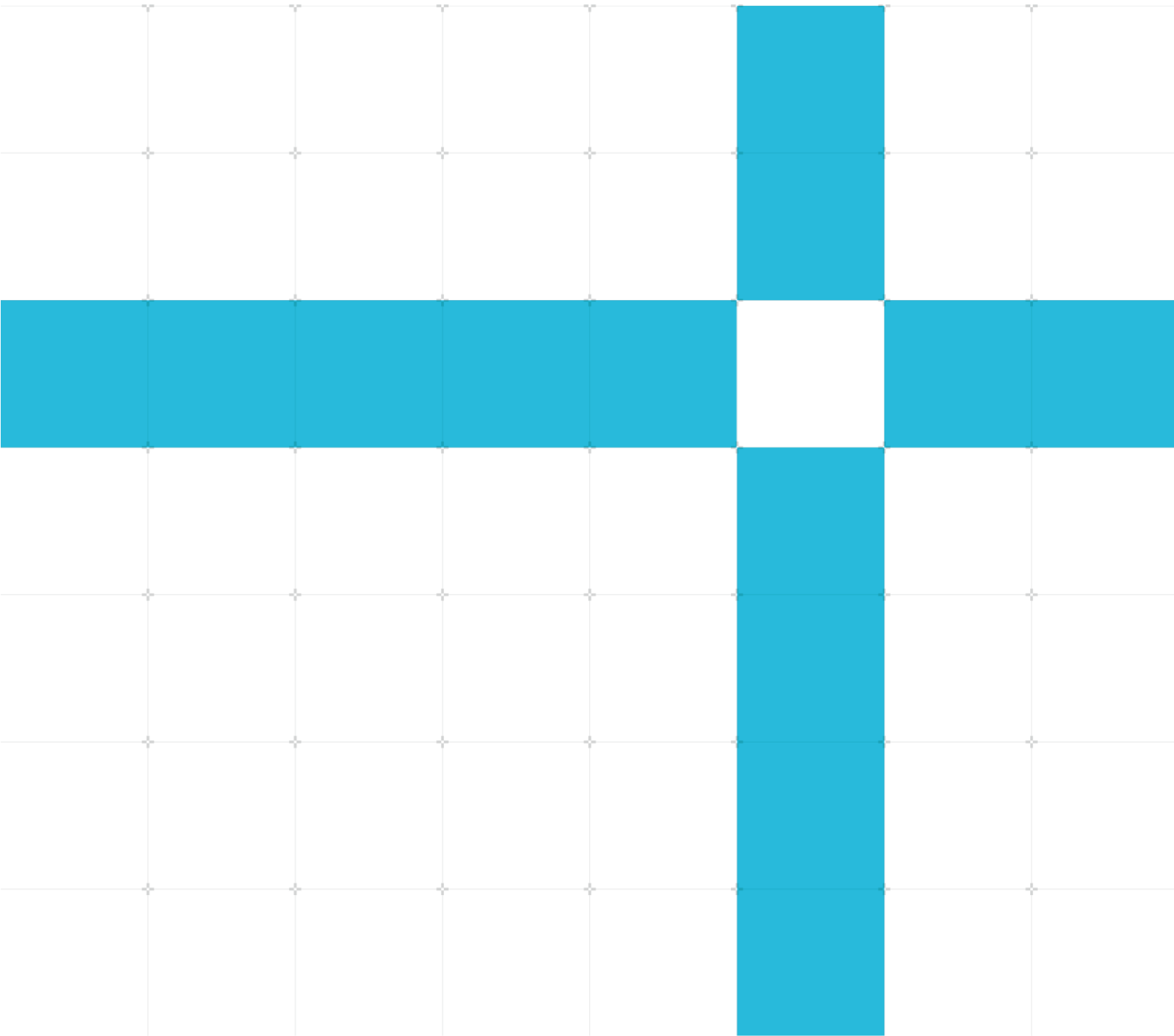
Product revision: r23p4-00rel0

Release Note

Non-Confidential

Issue 00

Copyright © 2023 Arm Limited (or its affiliates). DSHVE-DC-06003
All rights reserved.



Arm Mobile Studio 2023.4

Release Note

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.

Non-Confidential Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, has undertaken no analysis to identify or understand the scope and content of, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm's trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.
110 Fulfourn 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.....	5
1.1	Product description.....	5
1.1.1	Component versions	5
1.2	Release status.....	5
1.3	Changes in this release.....	5
1.3.1	Mobile Studio.....	5
1.3.2	Streamline.....	6
1.3.3	Graphics Analyzer	6
1.3.4	Mali Offline Compiler	6
1.4	Known issues in this release.....	6
1.4.1	Streamline.....	7
2	Support.....	8
2.1	How-to videos	8
2.2	Host OS support	8
2.3	Target OS support.....	8
2.4	Related projects.....	9
2.4.1	Mobile Studio for Unity package.....	9
2.4.2	ASTC Encoder texture compressor	9
2.4.3	HWCPipe library	10
2.4.4	libGPUInfo library	10
3	Installation.....	11
3.1	Install on Windows	11
3.2	Install on macOS	11
3.3	Install on Linux	12

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 8.8
- Graphics Analyzer 5.12.2
- Mali Offline Compiler 8.1

1.2 Release status

This is the REL quality release of the Arm Mobile Studio 2023.4 (r23p4-00rel0) software.

1.3 Changes in this release

This release of Arm Mobile Studio contains the following changes.

1.3.1 Mobile Studio

Mobile Studio has the following changes:

- Java runtime updated to Java 17.

1.3.2 Streamline

Streamline has the following changes:

- Android device connection now supports a simpler connection scheme. Simple mode automatically matches the Arm GPU counter template with the GPU present in the device. Manual configuration is still available for advanced users who need to use a non-default counter selection.
- Non-debuggable Android applications can now be profiled on rooted devices running **Eng** or **UserDebug** OS builds.
- Profiling with ArmNN machine learning library instrumentation is now supported for debuggable Android applications on devices running **User** OS builds.
- The Counter Configuration template menu now enables you to import custom counter templates.
- **Fix:** Debug information can now be loaded from stripped ELF images that contain empty **.text** and **.plt** sections.
- **Fix:** Arm GPU shader core performance counters now report the correct sum for devices with a sparse shader core mask.
- **Fix:** Immortalis-G715 and Mali-G715 series arithmetic unit utilization is now correctly computed in the Streamline template file.
- **Fix:** Command line exports of software Call Paths now correctly apply the **-thread** filter if specified.
- **Deprecation notice:** Energy profiling using Arm Energy Probe, or an NI DAQ probe, is a deprecated feature. It will be removed in a future release.

1.3.3 Graphics Analyzer

Graphics Analyzer has the following changes:

- Jython updated to version 2.7.2.

Graphics Analyzer has the following known issues:

- **SDDAP-12605:** Tool loading splash-screen renders upside down on macOS 14 (Sonoma).

1.3.4 Mali Offline Compiler

Mali Offline Compiler has the following changes:

- No update.

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-12605:** Tool loading splash-screen renders upside down on macOS 14 (Sonoma). Streamline toolbar menu is named “New Application” on macOS 14 (Sonoma).
- **SDDAP-12290:** The Mali DDK can fail to emit the Perfetto data required for the scheduling timeline visualization. This can result in entries with unidentified processes and queues. It can also result in time ranges which show as idle in the scheduler timeline when the GPU is clearly active in the counter data. This is fixed in the Mali r44p1 driver.
- **SDDAP-12592:** When using the light-weight interceptor graphics context FPS counters will show the last FPS value if the context is made idle without being destroyed.
- **SDDAP-11607:** macOS host installs do not show Arm disassembly views.
- **SDDAP-11426:** High DPI display scaling has been disabled by default on Linux hosts, due to persistent reliability issues across multiple distributions and graphics drivers. If desired, display scaling support can be re-enabled by setting the environment variable `STREAMLINE_ENABLE_HIDPI` to 1 and restarting the tool.

2 Support

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

- [Get started with Streamline](#)
- [Get started with Performance Advisor](#)
- [Get started with Graphics Analyzer](#)
- [Get started with Mali Offline Compiler](#)

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	20.04 (Focal Fossa) 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: 4.5.0 (June 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.

4.5.0 release changes:

- Improved compiler handling, reinstating invariant image output for builds that use Clang-14 or newer compilers.

- Improved universal binary support for macOS, supporting x86-64 SSE4.1 and AVX2, as well as Arm64 NEON from a single compressor binary.
- Improved performance using vectorized color endpoint packing.
- Improved image quality for LDR RGB and RGBA endpoints using a more exhaustive endpoint packing search.

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: 2.1.0 (August 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.

2.1.0 release changes:

- Added support for generation of user-friendly derived expressions for all supported GPUs.

The source code is available on GitHub:

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

2.4.4 libGPUInfo library

Current version: 1.0.0 (June 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.

1.0.0 release changes:

- Added an option for emitting YAML output to the command-line support utility.
- Added dynamic IP configuration query support for the Mali-G310 and Mali-G510 GPUs, as the arithmetic and texturing performance of each shader core can be configured by the chipset manufacturer.

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 3.8](#) (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 2023.4” shortcut,
- To launch Graphics Analyzer, open the Windows Start menu, navigate to the Arm Mobile Studio folder, and select the “Arm MS Graphics Analyzer 2023.4” 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_2023.4_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
```