



Arm Compiler for Linux

Version: 23.10

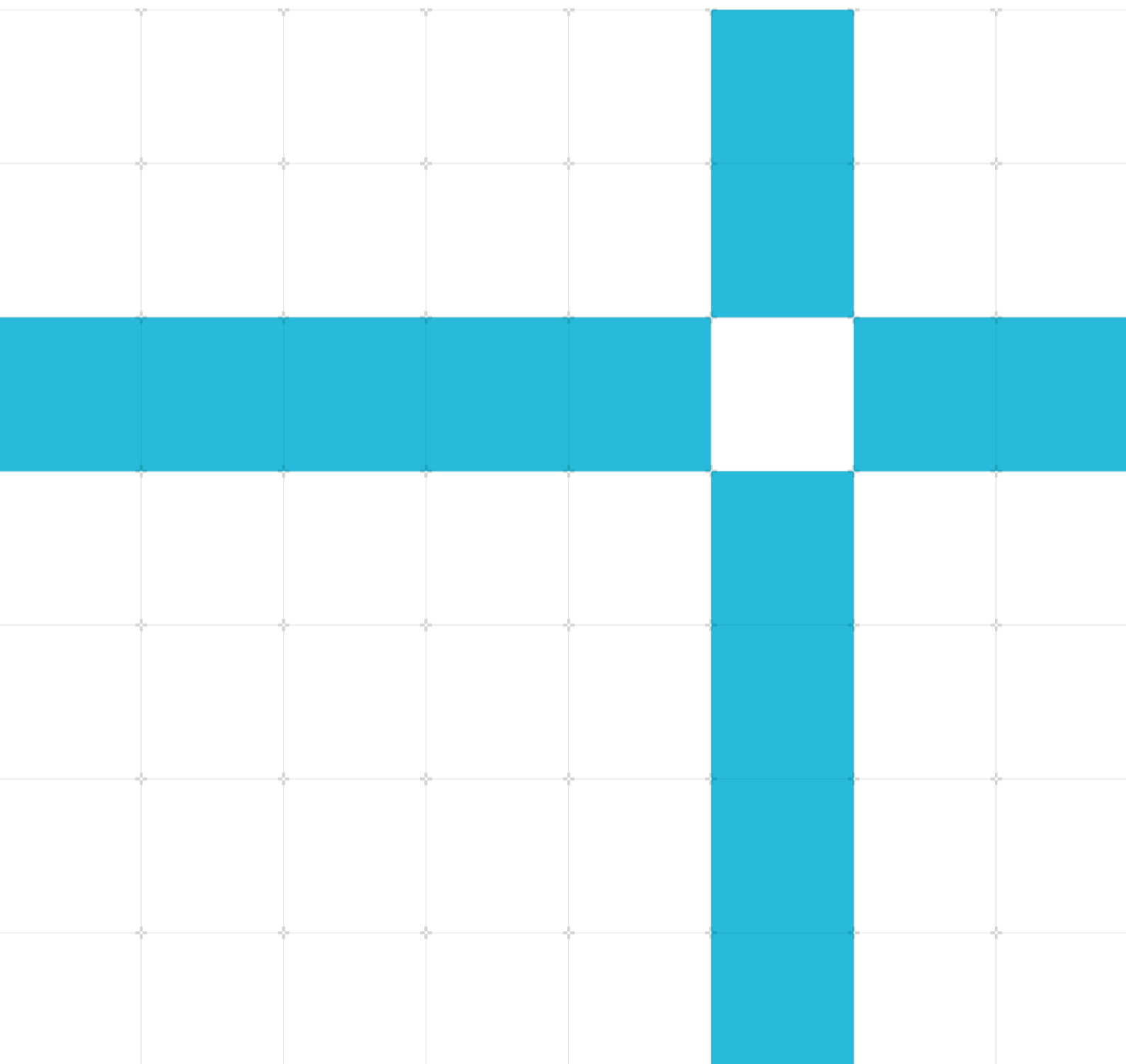
Release Note

Non-Confidential

Copyright © 2015-2023 Arm Limited (or its affiliates).
All rights reserved.

Issue 00

107578_2310_00



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, and 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 © [2015, 2023] Arm Limited (or its affiliates). All rights reserved.

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.

Product Status

The information in this document is Final, that is for a developed product.

Web Address

<https://developer.arm.com>

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.

We believe that this document contains no offensive terms. To report offensive language in this document, email terms@arm.com.

Contents

1	Release overview	1
1.1	Product description	1
1.2	Release status	1
2	Release contents	2
2.1	Deliverables	2
2.2	Differences from previous release	2
2.3	Known limitations	3
3	Get started	4
3.1	Licensing information	4
3.2	Prerequisites	4
3.3	Download the product	4
3.4	Unpack the product	4
3.5	Directory structure:	4
3.6	Install the product	4
3.7	RPM and DEB files	5
3.8	Run the product	5
3.9	Examples	6
3.10	Uninstall	7
4	Support	8
4.1	OS	8
5	Release history	9
6	Glossary	10

1 Release overview

1.1 Product description

The Arm® Compiler for Linux 23.10 suite provides a complete compiling environment for natively developing and tuning your server and HPC applications on Arm-based platforms.

The suite contains the following packages:

- Arm® C/C++/Fortran Compiler 23.10

Arm Compiler is a Linux user-space C/C++ and Fortran compiler tailored for scientific computing, HPC, and enterprise workloads.

- Arm® Performance Libraries 23.10.0

Arm® Performance Libraries contains optimized math functions, such as linear algebra and Fast Fourier Transforms, for Arm AArch64 implementations, including those with SVE. It is compatible with Arm® C/C++/Fortran Compiler 23.10 and GCC 12.2.0.

Arm® Performance Libraries is optimized for a number of microarchitectures. The latest information is available on the Arm Developer website:

<https://developer.arm.com/Tools%20and%20Software/Arm%20Compiler%20for%20Linux#Supported-Devices>

- GCC 12.2.0

For convenience, and to provide the optimal experience of using Arm Performance Libraries and GCC on the latest Arm server and HPC systems, a build of GCC 12.2.0 is provided. The GCC 12.2.0 build is also provided for OpenMP/libgfortran compatibility with Arm Performance Libraries.

As a GNU tool suite, the GPL-licensed source code can be downloaded separately.

1.2 Release status

This is the 23.10 release of the Arm® Compiler for Linux software.

These deliverables are being released under the terms of the agreement between Arm and each licensee (the “Agreement”). All planned verification and validation is complete. The release is suitable for volume production under the terms of the Agreement.

2 Release contents

The following subsections describe:

- The product parts that are delivered as part of this release.
- Any changes since the previous release.
- Any known issues and limitations that exist at the time of this release.

2.1 Deliverables

- Arm® C/C++/Fortran Compiler 23.10
- Arm® Performance Libraries 23.10.0
- GCC 12.2.0
- Release Notes (this document)
- Documentation

Arm® Compiler for Linux reference guides are available in `<install_location>/<package_name>/share`. The guides that are in the `'/share'` location are also available on the Arm Developer website:

<https://developer.arm.com/Tools%20and%20Software/Arm%20Compiler%20for%20Linux#Resources>

The same Arm Developer web page also contains links to tutorials, installation guides, and application porting guides.

Documentation and release notes might change between product releases. For the latest documentation bundle, check the product download page.

Arm tests PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its PDFs when used with any other PDF reader. Adobe reader products are available at <https://www.adobe.com>.

2.2 Differences from previous release

Arm® Compiler for Linux 23.10 includes various internal changes that resolve defects and improve performance.

The following subsections describe the significant differences from the previous release of Arm® Compiler for Linux.

Additions and changes:

This section describes the new features or components added, or any significant technical changes to features or components, in the 23.10 release.

- Arm® Compiler for Linux suite 23.10:
 - Added Support for SLES15 SP5
- Arm® C/C++/Fortran Compiler 23.10:
 - Arm® Compiler for Linux 23.10 has upgraded its underlying technology to LLVM version 17.0.0, which provides improved performance and stability. Please refer to the upstream release notes for more details:
 - * <https://releases.llvm.org/17.0.1/docs/ReleaseNotes.html>
 - The default setting of the compiler has changed to `-fno-math-errno` at `-O1` and above. In addition, `-fmath-errno` now implies `-fnosimdmath` as the vector math functions are not compatible with `-fmath-errno`.

- Arm® Performance Libraries 23.10.0:
 - The Arm Neoverse V2 and Neoverse N2 cores are added as new microarchitecture targets with specific tunings.
 - Increased performance for:
 - * Many cases across BLAS routines.
 - * Parallel FFT implementations.
 - New BLAS extension matrix-copy routines added:
 - * Out-of-place routines: ?OMATCOPY.
 - * In-place routines: ?IMATCOPY
 - * See examples and online documentation for details.

Resolved issues:

Describes any technical issues that are resolved in the 23.10 release.

- Arm® Compiler for Linux suite 23.10:
 - ACfL modules no longer set the ARM_LINUX_COMPILER_INCLUDES and ARM_LINUX_COMPILER_LIBRARIES environment variables, instead use ARM_LINUX_COMPILER_DIR/include and ARM_LINUX_COMPILER_DIR/lib respectively.
- Arm® C/C++/Fortran Compiler 23.10:
 - Fixed handling of the absolute paths in the #include directives used in the Fortran programs.
 - Clarified default setting for -fstack-arrays option in armflang -help message.
- Arm® Performance Libraries 23.10.0:
 - Arm® Performance Libraries now detects software availability of SVE on SVE capable hardware.

Describes any technical issues that are resolved in the 23.04.1 release.

- Arm® C/C++/Fortran Compiler 23.04.1:
 - Fixed a bug where armclang would crash with exit code 139 when compiling code using SVE types inside an OpenMP region with -fopenmp and targeting an SVE platform.
- Arm® Performance Libraries 23.04.1:
 - Integer overflow fixed in armpl_spmat_export* functions in lp64 libraries.

2.3 Known limitations

The following subsection describes any issues that are known at the time of this release.

Open technical issues:

Describes any technical issues that are open in the 23.10 release.

- Arm® Compiler for Linux suite 23.10:
 - In September 2023, a security vulnerability was disclosed on the GCC toolchain: CVE-2023-4039. The vulnerability can be exploited when using the stack protection feature of GCC. The GCC package distributed with ACfL does not contain the mitigations for this vulnerability. If you need to use the stack protection feature, please read the full update at: <https://developer.arm.com/Arm%20Security%20Center/GCC%20Stack%20Protector%20Vulnerability%20AArch64>

3 Get started

This section describes information to help you get started with accessing, setting up, and using Arm® Compiler for Linux.

For more information, see the Get Started information on the Arm Developer website:

<https://developer.arm.com/documentation/101458/2310/Get-started>

3.1 Licensing information

Use of Arm® Compiler for Linux is subject to the terms and conditions of the applicable End User License Agreement (“EULA”). A copy of the EULA can be found in the ‘license_terms’ folder of your product installation.

You do not require a license to use this Arm® Compiler for Linux package.

3.2 Prerequisites

If any of the following tools are not already installed by your Linux distribution, you must install them before installing Arm® Compiler for Linux. These packages can be installed with the appropriate package manager for your OS:

- SLES: awk environment-modules glibc-devel gzip python3 tar
- RHEL: environment-modules glibc-devel procps python3
- Amazon Linux: environment-modules glibc-devel gzip procps python3 tar
- Ubuntu: environment-modules libc6-dev python3

Note: The minimum supported version for Python is version 3.6.

You must have at least 2 GB of free hard disk space to both download and unpack the Arm® Compiler for Linux package. You must also have an additional 6 GB of free space to install the package.

3.3 Download the product

Arm delivers the files through the Arm Developer website:

[https://developer.arm.com/Tools%20and%20Software/Arm%20Compiler%20for%20Linux#
Technical-Specifications](https://developer.arm.com/Tools%20and%20Software/Arm%20Compiler%20for%20Linux#Technical-Specifications)

3.4 Unpack the product

To unpack the package, extract the tar file contents using a tar utility:

```
tar -xvf <package_name>.tar
```

3.5 Directory structure:

Shows the top-level directory structure of this installer package, which is available after you unpack the bundle:

```
license_terms/  
arm-compiler-for-linux-23.10*.sh  
RELEASE_NOTES.txt
```

3.6 Install the product

To install Arm® Compiler for Linux, navigate into the extracted package directory (<package_name>) and run the installation script as a privileged user. Pass any options to configure the installation:


```
cd path/to/<package_name>/  
./arm-compiler-for-linux-23.10*.sh [option]...
```

Some common installation options are:

- For a headless installation and to automatically accept the EULA, use the ‘-accept’ option.
- To install to an alternate location to the default, use the ‘-install-to <install_location>’ option.

For a full list of supported installation options pass the ‘-h’ or ‘-help’ options to the installer script.

To learn more about installing Arm® Compiler for Linux, see:

<https://developer.arm.com/documentation/102621/0100/Install-Arm-Compiler-for-Linux>

The installer displays the EULA and prompts you to agree to the terms. Type ‘yes’ at the prompt to continue.

All the packages are unpacked to <install_location>/<package_name> with environment modulefiles available under <install_location>/modulefiles. The default installation location is /opt/arm/. Local installs have the same directory structure starting from your chosen installation root.

3.7 RPM and DEB files

The install packages contain RPM (.rpm) files, for Linux distributions that use the Red Hat Package Manager (including SLES and Amazon Linux), or DEB (.deb) files, for Debian-based Linux distributions.

To extract the .rpm or .deb files from the installer, run the installer script with the ‘-s’ or ‘-save-packages-to <directory_location>’ option. If <directory_location> is not an empty directory, you also need to include the ‘-f’ or ‘-force’ option. The installer script requires you to accept the EULA. If you accept the EULA, the .rpm or .deb files extract to <directory_location>.

RPM files are signed by Arm’s HPC GPG key. DEB files are not signed. To verify RPM files, you can download and import the Arm’s HPC GPG key, and check the signatures:

1. Download the Arm HPC GPG public key from:

<https://developer.arm.com/-/media/files/keys/GPG-PUB-KEY-ARM-HPC-SW-TOOLS.PUB>

2. Import the GPG key, run:

```
rpm --import GPG-PUB-KEY-ARM-HPC-SW-TOOLS.PUB
```

3. Check the signature of an .rpm file, run:

```
rpm -K <rpm_file>
```

To install Arm® Compiler for Linux using rpm/deb files, navigate into the extracted rpm/deb files directory, run:

```
rpm -i <list_of_rpm_files>  
dpkg -i <list_of_deb_files>
```

Note: Arm does not recommend that you install directly from the .rpm or .deb files. Only experienced users who are comfortable with this type of installation route should attempt to install the Arm® Compiler for Linux package using this method.

3.8 Run the product

1. Load the environment module:

- Ensure you have access to modules, replace /opt/arm with <install_location> if necessary, and use:

```
module use /opt/arm/modulefiles  
module avail
```

- For Arm® C/C++/Fortran Compiler, use:

```
module load acfl/23.10
```

To also use Arm® Performance Libraries, include the `-armpl` compiler option when linking your executable. You do not need to load the Arm Performance Libraries modulefile.

- For GCC 12.2.0 only, use:

```
module load gnu/12.2.0
```

- For GCC 12.2.0 with Arm® Performance Libraries, use:

```
module load gnu/12.2.0  
module load armpl/23.10.0
```

2. Generate your executable binary.

To generate an executable binary with Arm® Compiler for Linux, compile your program with Arm® C/C++/Fortran Compiler and specify any options ([options]), the output binary name (`-o <binary>`), and the input file (`<input>`):

```
{armclang|armclang++|armflang} [options] -o <binary> <input>
```

Refer to the GCC documentation to see the equivalent command syntax for the GCC compiler.

3.9 Examples

Example code is included in this suite as part of Arm Performance Libraries. This code can be found at:

```
<install_location>/<ARMPL_Name>*<ARMPL_Version>*/examples*
```

Examples that use, and do not use, SVE are included for each of Arm C/C++/Fortran Compiler and GCC.

Multiple examples directories are provided in the installation. The suffix of the directory name indicates whether the examples inside link to the 32-bit ('_lp64') or 64-bit ('_ilp64') integer variants, and sequential (no suffix indicator) or OpenMP ('_mp') multi-threaded variants, of Arm Performance Libraries.

The default set of examples in the 'examples' directory link to the sequential, 32-bit integers variant of Arm® Performance Libraries.

To build the default set of examples:

- For Arm® Compiler for Linux:

1. Copy the 'examples' directory somewhere writeable:

```
cp -r <install_location>/armpl-23.10.0*arm-linux-compiler/examples ./  
cd examples
```

2. Load the Arm® Compiler for Linux environment module:

```
module load acfl/23.10
```

3. Build the examples:

```
make
```

- For GCC:

1. Copy the 'examples' directory somewhere writeable:

```
cp -r <install_location>/armpl-23.10.0*gcc/examples ./  
cd examples
```

2. Load the GCC environment modules:

```
module load gnu/12.2.0  
module load armpl/23.10.0
```

3. Build the examples:

```
make
```

- Note: The examples above only work with the modules listed in their respective step 2. For instance, the GCC examples will not work if the Arm® Compiler for Linux variant of the performance libraries is already loaded. The modules environment can be cleaned with “module purge”
- Note: For Linux distributions that use the Red Hat Package Manager, if you want to change the install location of ACfL, you need to first uninstall the package and then re-install it to the desired location. This is only required when installing the product as a privileged user.

For more information about the Arm® Performance Libraries examples, see:

<https://developer.arm.com/documentation/102574/0100/Compile-and-test-the-examples>

3.10 Uninstall

For convenience, this package includes an “uninstall.sh” script at:

```
<install_location>/arm-compiler-for-linux-23.10*/uninstall.sh
```

This script attempts to uninstall all the components supplied as part of Arm® Compiler for Linux. However, if other packages outside of this product depend on the GCC component, GCC will not be uninstalled. Packages extracted using the `-save-packages-to` option and installed using `rpm/dpkg` commands should be uninstalled in the same way and deleted manually.

4 Support

The documentation that is available for Arm® Compiler for Linux can be found on the product resources page on the Arm Developer website:

<https://developer.arm.com/Tools%20and%20Software/Arm%20Compiler%20for%20Linux#Resources>

You can also find a subset of that documentation, available in

`<install_location>/<package_name>/share.`

These deliverables are being released under the terms of the agreement between Arm and each licensee (the “Agreement”). All planned verification and validation is complete. The release is suitable for volume production under the terms of the Agreement.

4.1 OS

This suite is supported on the following Linux platforms:

- AArch64 RHEL 7, 8 and 9
- AArch64 SLES 15
- AArch64 Ubuntu 20.04 and 22.04
- AArch64 Amazon Linux 2 and 2023

Full information about the platforms supported by Arm® Compiler for Linux is available on the Arm Developer website:

<https://developer.arm.com/Tools%20and%20Software/Arm%20Compiler%20for%20Linux#Supported-Devices>

5 Release history

A full release history (with release notes) for Arm® Compiler for Linux is available on the Arm Developer website:

<https://developer.arm.com/documentation/101458/latest>

6 Glossary

The Arm Glossary is a list of terms that are used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: <https://developer.arm.com/glossary>.