



Release Notes for Arm Compiler for Embedded FuSa 6.22.1

Version 6.22.1

Non-Confidential

Copyright © 2024 Arm Limited (or its affiliates).
All rights reserved.

Issue 00

109905_062201_00_en



Release Notes for Arm Compiler for Embedded FuSa 6.22.1

This document is Non-Confidential.

Copyright © 2024 Arm Limited (or its affiliates). All rights reserved.

This document is protected by copyright and other intellectual property rights.

Arm only permits use of this document if you have reviewed and accepted [Arm's Proprietary Notice](#) found at the end of this document.

This document (109905_062201_00_en) was issued on 2024-08-21. There might be a later issue at <https://developer.arm.com/documentation/109905>

The product version is 6.22.1.

See also: [Proprietary notice](#) | [Product and document information](#) | [Useful resources](#)

Start reading

If you prefer, you can skip to [the start of the content](#).

Intended audience

This document is intended for use by a software developer who is using Arm Compiler for Embedded FuSa to build a project with functional safety or long-term maintenance requirements. The document includes an overview of the Arm Compiler for Embedded FuSa 6.22.1 release, changes and enhancements, and defects fixed.

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

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, 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>.

Contents

1. Release overview.....	4
1.1 Product description.....	4
1.2 Release highlights.....	5
1.3 Included components.....	7
1.4 Product quality.....	8
2. Download Arm Compiler for Embedded FuSa 6.22.1.....	9
3. Differences from previous release.....	11
3.1 Defect fixes.....	11
3.1.1 Compiler and integrated assembler, armclang.....	11
3.1.2 Linker, armlink.....	12
3.1.3 Libraries and system headers.....	12
3.2 Deprecated or removed features.....	12
4. Support.....	13
5. Release history.....	14
Proprietary notice.....	15
Product and document information.....	17
Product status.....	17
Revision history.....	17
Conventions.....	17
Useful resources.....	20

1. Release overview

This chapter provides an overview of the Arm Compiler for Embedded FuSa product and the Arm Compiler for Embedded FuSa 6.22.1 release.

1.1 Product description

Arm Compiler for Embedded FuSa is the most advanced embedded C/C++ compilation toolchain from Arm for the development of bare-metal software, firmware, and Real-Time Operating System (RTOS) applications with functional safety (FuSa) or long-term support requirements.

The toolchain is compatible with the following Arm Integrated Development Environments (IDEs):

- Arm Development Studio
- Arm Keil MDK v6

Through powerful optimization techniques and optimized libraries, Arm Compiler for Embedded enables functional safety embedded system developers to meet challenging performance goals and memory constraints.

Arm Compiler for Embedded FuSa is used by leading companies in a wide variety of industries, including automotive (ISO 26262), consumer electronics, industrial (IEC 61508), medical (IEC 62304), networking, railway (EN 50716), storage, and telecommunications.

Key changes in Arm Compiler for Embedded FuSa 6.22LTS when compared to the first release of Arm Compiler for Embedded FuSa 6.16LTS include:

- Support for the following Automotive Enhanced processors:
 - Neoverse V3AE
 - Cortex-A720AE, Cortex-A520AE
 - Cortex-R82AE
- Support for the following processors:
 - Neoverse V2
 - Cortex-X4, Cortex-X3, Cortex-X2, Cortex-X1C
 - Cortex-A720, Cortex-A715, Cortex-A710, Cortex-A520, Cortex-A510
 - Cortex-R82
 - Cortex-M85, Cortex-M52
- Support for the following architectures:
 - Armv9-A up to Armv9.5-A
 - Armv8.8-A and Armv8.9-A
 - Armv8-R AArch64 implementations without hardware floating-point, except when compiling in a C++ source language mode

- Armv8-R AArch64 implementations with hardware floating-point
- Support for the following architecture extensions:
 - 2023 extensions for A-profile architectures
 - The M-profile PACBTI Extension
- Support for User-based Licensing (UBL)
- Removal of FlexNet Publisher (FNP) and Keil licensing
- Support for C++17
- Support for execute-only memory (XOM) regions when building for an Armv6-M target
- Support for the following security features:
 - Undefined Behavior Sanitizer (UBSan)
 - Control Flow Integrity (CFI)
 - Shadow Call Stack for AArch64 state
 - Mitigations for the PACMAN security vulnerability
 - Mitigations for the Straight-Line Speculation (SLS) security vulnerability
 - Mitigations for the VLLDM instruction security vulnerability
- Support for automatic vectorization for SVE and SVE2
- Deprecation of the legacy assembler, armasm

For a summary of changes between Arm Compiler for Embedded FuSa 6.16LTS and Arm Compiler for Embedded FuSa 6.22LTS, see the *Summary of changes between Arm Compiler for Embedded FuSa 6.16LTS and Arm Compiler for Embedded FuSa 6.22LTS* section of the [Arm Compiler for Embedded FuSa Migration and Compatibility Guide](#) document for this release.

Contact your sales representative or [submit an inquiry online](#) to find out more about licensing Arm Compiler for Embedded FuSa or an Arm IDE.

1.2 Release highlights

Arm Compiler for Embedded FuSa 6.22.1 is the latest release as of August 2024. It has been derived from the unqualified Arm Compiler for Embedded 6.22 release, includes additional defect fixes, and adds a Qualification Kit for functional safety purposes.

Subject to your license terms, Arm Compiler for Embedded FuSa 6.22.1 can be used to build for the following Arm Architectures and Processors:

Architecture	Processor Family	Standard Processors	Automotive Enhanced Processors
Armv9-A up to Armv9.5-A	Neoverse	V2	V3AE
		N2	

Architecture	Processor Family	Standard Processors	Automotive Enhanced Processors
	Cortex	X4, X3, X2 A720, A715, A710 A520, A510	A720AE, A520AE
Armv8-A up to Armv8.9-A	Neoverse	V1 N1 E1	-
	Cortex	X1C, X1 A78C, A78, A77, A76, A75, A73, A72 A65 A57, A55, A53 A35, A34, A32	A78AE, A76AE A65AE
Armv7-A	Cortex	A17, A15, A12 A9, A8, A7, A5	-
Armv8-R AArch64	Cortex	R82	R82AE
Armv8-R	Cortex	R52+, R52	-
Armv7-R	Cortex	R8, R7, R5, R4F, R4	-
Armv8-M up to Armv8.1-M	Cortex	M85 M55, M52 M35P, M33 M23	-
	STAR	STAR-MC1	-
Armv7-M	Cortex	M7, M4, M3	-
	SecurCore	SC300	-
Armv6-M	Cortex	M1, M0, M0+	-
	SecurCore	SC000	-

For more information, see the following:

- The [Arm Development Studio](#) product page.
- The [Arm Keil MDK v6](#) product page.
- The *Support level definitions* section of the [Arm Compiler for Embedded FuSa User Guide](#) document for this release.

1.3 Included components

This section lists the toolchain components and different types of documentation included in Arm Compiler for Embedded FuSa 6.22.1.

Category	Component	Description
Qualified toolchain components	<code>armclang</code>	Compiler and integrated assembler based on LLVM and Clang technology
	<code>armar</code>	Archiver which enables sets of ELF object files to be collected together
	<code>armlink</code>	Linker that combines objects and libraries to produce an executable
	<code>fromelf</code>	ELF image conversion utility and disassembler
Unqualified toolchain components	Arm C libraries	Runtime support libraries for embedded systems
	Arm C++ libraries	Libraries based on the LLVM libc++ project
	<code>armasm</code>	Deprecated legacy assembler for <code>armasm-syntax</code> assembly code for older Arm architectures only. Use the <code>armclang</code> integrated assembler for all new assembly files.
Qualification Kit	Safety Manual	Describes the scope of qualification, and how to use the toolchain for safety-related development
	Defect Report	Provides information about known safety-related defects at the time of release
	Test Report	Contains results from language conformance tests
	Development Process	Contains an overview of the process used to develop the toolchain
	Release History	Identification information for all the releases to date of the Arm Compiler for Embedded FuSa 6.22LTS release series
User documentation	User Guide	Provides instructions and examples to help you use the toolchain
	Reference Guide	Provides information to help you configure the toolchain
	Arm C and C++ Libraries and Floating-Point Support User Guide	Provides information about the Arm libraries and floating-point support
	Errors and Warnings Reference Guide	Provides a list of the errors and warnings that can be reported by <code>armar</code> , <code>armasm</code> , <code>armlink</code> , and <code>fromelf</code>
	Migration and Compatibility Guide	Provides information to help you migrate from Arm Compiler 5 to Arm Compiler for Embedded FuSa
	Release Notes	These release notes

These components can be obtained from the following sources:

Component type	Source
Qualified toolchain components	Available via Product Download Hub (PDH)
Unqualified toolchain components	Available via PDH
Qualification Kit	Available via PDH
Documentation	Available via Arm Developer

1.4 Product quality

This product is a Final release quality product which is suitable for use in a production environment.

Certain features within this product are not Final release quality features or are unsupported. The status of features is explicitly stated within the documentation where applicable. For more information about such features, see the the following:

- The *Features not recommended for use in safety-related development* section of the *Arm Compiler for Embedded FuSa Qualification Kit Safety Manual*.
- The *Support level definitions* section of the [Arm Compiler for Embedded FuSa User Guide](#).

2. Download Arm Compiler for Embedded FuSa 6.22.1

This chapter provides information about how to download and install Arm Compiler for Embedded FuSa 6.22.1.

Arm Compiler for Embedded FuSa 6.22.1 might be available to download standalone, as part of an Arm Integrated Development Environment (IDE), or as part of a Success Kit depending on your license and entitlements.

To download this release standalone, use the `ACOMP622` code on Arm Product Download Hub (PDH). The toolchain download packages within this product code are intended to be used in the following environments as of August 2024^{*}:

Host architecture	Host operating system	Toolchain download package	Host environment
x86_64	Red Hat Enterprise Linux 9	x86_64 Linux	Standalone installation
	Red Hat Enterprise Linux 8		Integrated into Arm Development Studio
	Red Hat Enterprise Linux 7		Integrated into Keil MDK version 6
	Ubuntu 22.04 LTS		
	Ubuntu 20.04 LTS		
	Windows Server 2022	x86_64 Windows	Standalone installation
	Windows Server 2019		Integrated into Arm Development Studio
	Windows 11		Integrated into Keil MDK version 6
	Windows 10		
AArch64	Ubuntu 20.04 LTS	AArch64 Linux	Standalone installation
			Integrated into Keil MDK version 6

^{*} Arm reserves the right to add additional environments for this release.

The Qualification Kit documents can be obtained from additional download packages in PDH under the same product code, `ACOMP622`.

The following restrictions apply:

- The minimum required version of glibc for Linux host platforms is as follows:
 - 2.15 for x86_64 host platforms.
 - 2.17 for AArch64 host platforms.
- The toolchain must not be installed directly into an Arm Development Studio installation directory.

For more information, see the following:

- The *System requirements and installation* section of the [Arm Compiler for Embedded FuSa User Guide](#) for toolchain installation instructions.
- The *Register a compiler toolchain* section of the [Arm Development Studio Getting Started Guide](#) for instructions to integrate the toolchain into Arm Development Studio.
- The [Arm Keil Studio Visual Studio Code Extensions User Guide](#) for instructions to integrate the toolchain into [Arm Keil MDK v6](#).
- The [User-based Licensing User Guide](#) for instructions to configure the toolchain to use a User-based License (UBL).

3. Differences from previous release

This chapter describes differences from the previous unqualified feature release, Arm Compiler for Embedded 6.22.

For more information about the scope of this section, see the article [Does Arm document all known issues that affect each Arm Compiler release?](#).

The information below may include technical inaccuracies or typographical errors. Each itemized change is accompanied by a unique SDCOMP-*<n>* identifier. If you need to contact Arm about a specific issue within these release notes, please quote the appropriate identifier.

3.1 Defect fixes

This section contains information about defect fixes made in this release of Arm Compiler for Embedded FuSa. It contains sub-sections that each focus on the defect fixes in a specific component of the toolchain.

3.1.1 Compiler and integrated assembler, armclang

This section contains a list of defect fixes made in the compiler and integrated assembler, armclang.

- [SDCOMP-65172] When compiling with `-gdwarf-2` or `-gdwarf-3`, the compiler could generate incorrect debug information for a bit-field. This has been fixed.
- [SDCOMP-65141] When assembling for a big-endian target, the inline assembler and integrated assembler generated incorrect code for a sequence of `LDR` instructions that loads the same literal value into a combination of 32-bit and 64-bit registers. This has been fixed.
- [SDCOMP-65094] When compiling without `-fno-optimize-sibling-calls`, with an `-mbranch-protection=<protection>` option that enables Pointer Authentication Code (PAC) protection, and for an Armv8.1-M target with the Main Extension, the compiler could generate incorrect code for a function that uses `R12` for a named register variable. This has been fixed.
- [SDCOMP-64829] The compiler could incorrectly generate an architecturally `CONSTRAINED UNPREDICTABLE` instruction for a target that supports the Standardization of memory operations feature (FEAT_MOPS). This has been fixed.
- [SDCOMP-63913] When compiling with a `-mharden-sls=<option>` option that enables the mitigation against Straight-Line Speculation (SLS) for `RET` and `BR` instructions, and with `-moutline` or at `-Oz` without `-mno-outline`, the compiler could generate incorrect code. This has been fixed.

3.1.2 Linker, armlink

This section contains a list of defect fixes made in the linker, `armlink`.

- [SDCOMP-65517] For two execution regions `A` and `B`, where `A` is at a lower address than `B`, the linker incorrectly assumed that the name of `A` is always lexically earlier than the name of `B`. Subsequently, this could result in the linker generating incorrect callgraph or stack usage information. This has been fixed. The linker no longer assumes that the name of `A` is always lexically earlier than the name of `B`.
- [SDCOMP-64999] When linking an input object that has been compiled with C++ exceptions enabled and for AArch32 state, the linker could generate an incorrect C++ exception-handling table. This has been fixed.

3.1.3 Libraries and system headers

This section contains a list of defect fixes made in the unqualified C and C++ libraries and system headers supplied with the toolchain.

- [SDCOMP-65388] The Arm C library was not thread-safe for a multithreaded program that contains C++ objects with static storage duration. This has been fixed.

3.2 Deprecated or removed features

This section contains a discretionary list of features that have been deprecated or removed in this release.

- [SDCOMP-64971] Support has been removed for FlexNet Publisher (FNP) licensing and Keil licensing. The toolchain only supports User-based Licensing (UBL).

For more information about UBL, see the [User-based Licensing User Guide](#).

4. Support

This chapter includes guidance on how to obtain support for using Arm Compiler for Embedded FuSa 6.22.1.

Your feedback is important to us, and you are welcome to send us defect reports and suggestions for improvement on any aspect of the product. Please contact your supplier or [open a case](#) with feedback or support issues, using your work or academic email address if possible. Where appropriate, please provide the following information:

- `--vsn` output from the tool.
- The complete content of any error message that the tool produces.
- Preprocessed source code, other files, and command-line options necessary to reproduce the issue. For information on how to preprocess source code, see the `-E` section of the [Arm Compiler for Embedded FuSa Reference Guide](#).

5. Release history

This chapter contains the history of Arm Compiler for Embedded FuSa 6.22LTS releases.

Version	Release Date
Arm Compiler for Embedded FuSa 6.22.1	21 Aug 2024

Proprietary Notice

This document is protected by copyright and other related rights and the use 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 Limited ("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 the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

This document may include technical inaccuracies or typographical errors. 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, any patents, copyrights, trade secrets, trademarks, or other rights.

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.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted 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 this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

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. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in Arm documents.

Product status

All products and services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

Product completeness status

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

Revision history

These sections can help you understand how the document has changed over time.

Document release information

The Document history table gives the issue number and the released date for each released issue of this document.

Document history

Issue	Date	Confidentiality	Change
062201-00	21 August 2024	Non-Confidential	Initial release

Change history

This section lists the changes made in each issue of this release notes document. It does not cover technical changes made in the product Arm Compiler for Embedded FuSa 6.22.1 itself.

Conventions

The following subsections describe conventions used in Arm documents.

Glossary

The Arm Glossary is a list of terms 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: developer.arm.com/glossary.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
<i>italic</i>	Citations.
bold	Interface elements, such as menu names. Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace <u>underline</u>	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: <div>MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2></div>
SMALL CAPITALS	Terms that have specific technical meanings as defined in the Arm® Glossary. For example, IMPLEMENTATION DEFINED , IMPLEMENTATION SPECIFIC , UNKNOWN , and UNPREDICTABLE .



We recommend the following. If you do not follow these recommendations your system might not work.



Your system requires the following. If you do not follow these requirements your system will not work.



You are at risk of causing permanent damage to your system or your equipment, or harming yourself.



This information is important and needs your attention.



A useful tip that might make it easier, better or faster to perform a task.



A reminder of something important that relates to the information you are reading.

Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at developer.arm.com/documentation. Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
Arm Compiler for Embedded FuSa 6.22LTS documentation index	KA006002	Non-Confidential
Arm Compiler for Embedded FuSa Migration and Compatibility Guide	109444	Non-Confidential
Arm Compiler for Embedded FuSa Qualification Kit Safety Manual	109409	Confidential
Arm Compiler for Embedded FuSa Reference Guide	109443	Non-Confidential
Arm Compiler for Embedded FuSa User Guide	109442	Non-Confidential
Arm Development Studio	–	Non-Confidential
Arm Development Studio Getting Started Guide	101469	Non-Confidential
Arm Keil MDK v6	–	Non-Confidential
Arm Keil Studio Visual Studio Code Extensions User Guide	108029	Non-Confidential
Does Arm document all known issues that affect each Arm Compiler release?	KA005052	Non-Confidential
User-based Licensing User Guide	102516	Non-Confidential