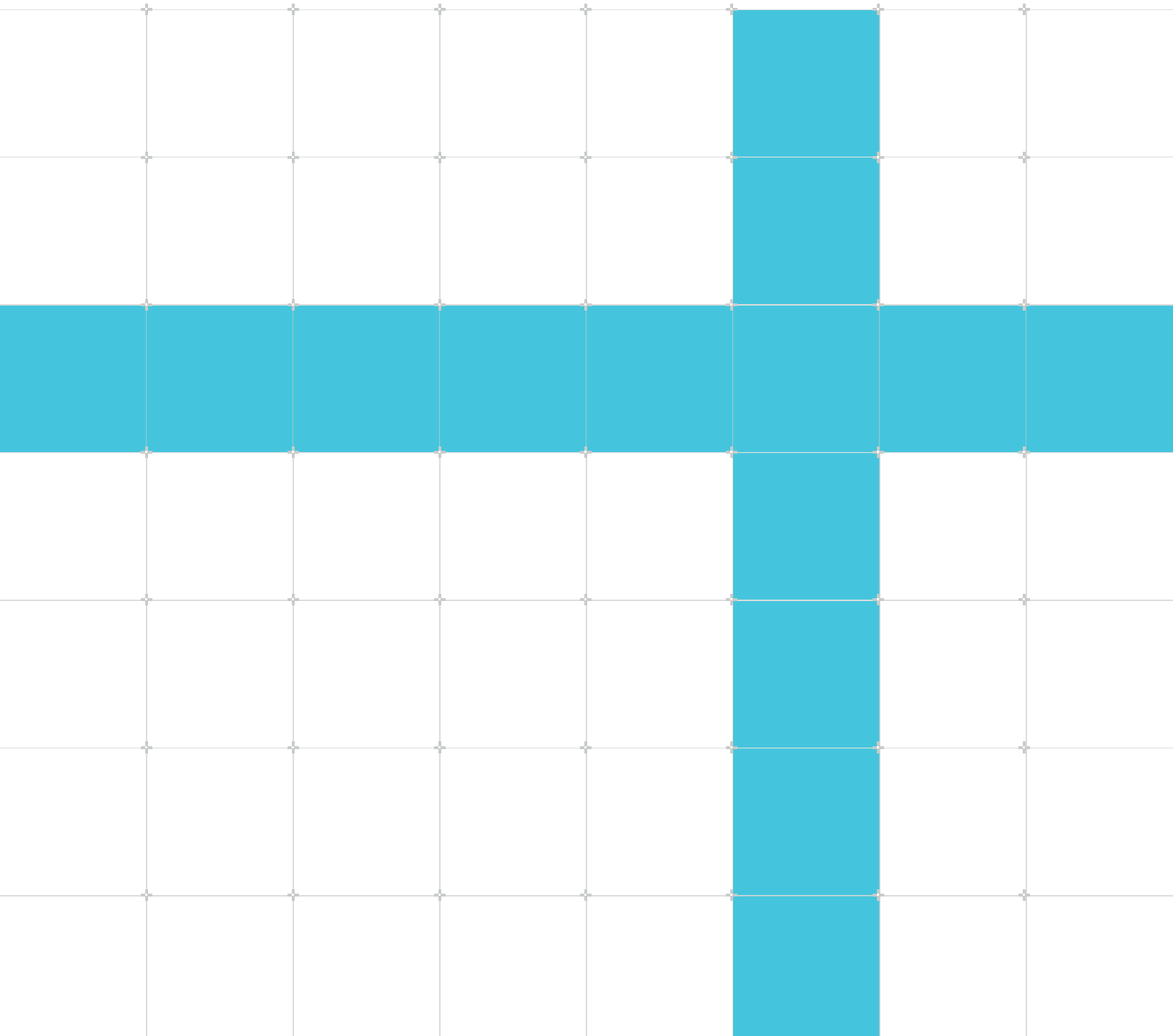
Arm® BSA Architecture Compliance

Version v1.2.0

User Guide

Issue 01
102504_0102_01_en



Arm® BSA Architecture Compliance User Guide

This document (102504_0102_01_en) was issued on 2025-12-10. There might be a later issue at https://developer.arm.com/documentation/ 102504

The product version is v1.2.0.

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 book is written for engineers who are designing or verifying an implementation of the Arm® Base System Architecture.

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.

This document includes language that can be offensive. We will replace this language in a future issue of this document.

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. Overview

This chapter provides an overview of the BSA tests and the test IDs.

1.1 Abbreviations

This section lists the abbreviations used in this document.

Table 1-1: Abbreviations and expansions

Abbreviation	Expansion
ACPI	Advanced Configuration and Power Interface
BSA	Base System Architecture
DT	Device Tree
GIC	Generic Interrupt Controller
HVC	HyperVisor Call
PAL	Platform Abstraction Layer
PCIe	Peripheral Component Interconnect express
PE	Processing Element
PSCI	Power State Coordination
SMC	Secure Monitor Call
SMMU	System Memory Management Unit
UEFI	Unified Extensible Firmware Interface

1.2 Overview of test suites

The following table describes the general divisions of Base System Architecture (BSA) tests between Unified Extensible Firmware Interface (UEFI) shell application and Linux application.

Table 1-2: Test environment and modules

Test environment	Modules
UEFI shell	PE, GIC, Timers, Watchdog, Wakeup and Power, PCIe, Memory map, Exerciser, Peripheral, and SMMU
Linux command line	PCIe, Memory map, and Peripheral

2. UEFI shell application

This chapter provides information on executing tests from the UEFI shell application and the PAL API implementation.

2.1 UEFI shell application arguments

This section provides information on the UEFI shell application arguments.

Run the UEFI shell application with the following set of arguments:

```
uefi shell> Bsa.efi [-v <n>] [-f <filename>] [-dtb <file>] [-os] [-hyp] [-ps] [-
l <n>] [-only <n>] [-r <RuleIDs|rules.txt>][-skip <RuleIDs>] [-m <modules>] [-
skipmodule <modules>] [-mmio] [-no_crypto_ext] [-p2p] [-cache] [-el1physkip] [-skip-
dp-nic-ms] -timeout <n>] [-h|-help] [-fr]
```

Note

The shell session becomes unusable after all the BSA tests are run and the test results are printed on the UEFI console.

The following table provides descriptions to the arguments.

Table 2-1: Description of UEFI application arguments

Argument	Description
`-cache`	Pass this flag to indicate that the system supports PCIe Address Translation Cache (ATC).
`-dtb <file>`	Saves the board Device Tree (DT) blob to the specified file.
`-el1physkip`	Skips EL1 register checks. Use this option for debug purposes in VE or EL1 only scenarios.
`-f <filename>`	File name to which the output log is written.
`-fr`	Runs rules up to the Future Requirements (FR) level.
`-h, -help`	Prints the help message and exits.
`-l`	Level of compliance to be tested for. The default value is 1.
`-m <modules>`	Runs only the specified modules. Provide a comma-separated list. Accepted values: PE, GIC, PERIPHERAL, MEM_MAP, PMU, RAS, SMMU, TIMER, WATCHDOG, NIST, PCIE, MPAM, ETE, TPM, and POWER_WAKEUP.
`-mmio`	To enable pal_mmio_read or write prints, use with -v 1.
`-no_crypto_ext`	Use this when cryptography extensions are unavailable due to export restrictions.
`-only <n>`	Runs only the tests for level n.

Argument	Description

`-os`
 	
`-hyp`
 	
`-ps`
	By default, all the operating system, hypervisor, and platform security view tests are run.
	
	To run specific tests, add the following options:
	
	`-os` Run the operating system view tests.
	`-hyp` Run the hypervisor view tests.
	`-ps` Run the platform security view tests.

`-p2p`	Pass this flag to indicate system support PCIe p2p.
`-r <RuleIDs|rules.txt>`	Runs only the specified rules. You can provide a comma-separated list of Rule IDs (for example: `B_PE_01`, `B_GIC_01`) or a rules file. The rules file may contain comma-separated or newline-separated Rule IDs. Lines that begin with # are treated as comments and ignored. **Note:** Rule ID string passed must match the BSA specification.
`-skip <RuleIDs>`	Skips the specified rules. Provide a comma-separated list of Rule IDs, using the same format as the `-r` option. Example: `-skip B_PE_01,B_GIC_02`
`-skip-dp-nic-ms`	Skips PCIe tests for DisplayPort, network, and mass storage devices.
`-skipmodule <modules>`	Skips the specified modules. Provide a comma-separated list, using the same format as the `-m` option.
`-timeout <n>`	Timeout value for the wakeup test. **1** Minimum value. **5** Maximum value. **1** Default.
`-v`	Print level **1** INFO and above. **2** DEBUG and above. **3** TEST and above. **4** WARN and ERROR. **5** ERROR.

Note

`-dtb` option is for platforms that present DT files only.

Examples

The following examples show how to run the UEFI shell application using arguments:

```
shell > Bsa.efi -v 2 -skip B_PE_01,B_SMMU_01 -skipmodule GIC,TIMER -f acs.txt -os -
dtb platform.dtb
```

The set of parameters shown in the code block:

- Prints messages with verbosity of 2 and above.
- Tests for compliance against operating system view tests.
- Skips execution of all rules belonging to Generic Interrupt Controller (GIC), Timer modules and tests for rule IDs B_PE_01 and B_SMMU_01.
- Stores the log messages to the `acs.txt` file.
- Saves the firmware DT into the `platform.dtb` file.

```
shell > bsa.efi -m PCIE -skip B_PCIe_10
```

The set of parameters shown in the code block:

- Runs only PCIE module rules.
- Skips PCIE rule B_PCIe_10.

2.2 UEFI shell implementation of PAL APIs

This section provides information on infrastructure APIs and module-specific APIs.

Booting to a UEFI shell is a prerequisite for running a BSA test.

Infrastructure APIs

The following table describes the Platform Abstraction Layer (PAL) APIs and UEFI interfaces.

Table 2-2: PAL APIs and UEFI interfaces

PAL API	UEFI interface
`pal_print`	AsciiPrint
`mem_alloc`	gBS->AllocatePool
`mem_free`	gBS->FreePool
`mem_alloc_shared`	gBS->AllocatePool
`mem_free_shared`	gBS->FreePool
`aligned_alloc`	gBS->AllocatePool

PAL API	UEFI interface
mem_free_aligned	gBS->FreePool
mem_get_shared_addr	None
mmio_read	None
mmio_write	None

Module-specific APIs

The following table represents the mapping of PAL API to Advanced Configuration and Power Interface (ACPI), if the system firmware presents platform configuration through ACPI tables.

Table 2-3: PAL APIs, UEFI interfaces, and ACPI tables consumed

PAL APIs	UEFI interfaces	ACPI tables consumed
pe_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi61.h	MADT Table
call_smc	-	-
pe_execute_payload	-	-
pe_install_esr	• gEfiCpuArchProtocolGuid • Cpu->RegisterInterruptHandler	-
gic_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi61.h	MADT table
gic_install_isr	• gHardwareInterruptProtocolGuid • RegisterInterruptSource • EnableInterruptSource	-
timer_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi61.h	GTDT table
wd_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi61.h	GTDT table
pcie_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi61.h	MCFG table
pcie_get_mcfg_ecam	• gST->ConfigurationTable • CompareGuid, IndustryStandard/Acpi61.h • IndustryStandard/ MemoryMappedConfigurationSpaceAccessTable.h	MCFG table
iovirt_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi61.h	IORT table

PAL APIs	UEFI interfaces	ACPI tables consumed
`peripheral_ create_info_table`	• gEfiPciIoProtocolGuid • Pci->GetLocation • Pci->Pci.Read	-
`memory_create_info_table`	gBS->GetMemoryMap	-

The following table represents the mapping of PAL API to DT node, if the system firmware presents platform configuration through DT nodes.

Table 2-4: PAL APIs, UEFI interfaces, and DT nodes consumed

PAL APIs	UEFI interfaces	DT nodes consumed
`pe_create_info_table`	• gST->ConfigurationTable • CompareGuid	cpu, pmu, interrupt-controller node
`gic_create_info_table`		interrupt-controller, v2m and its nodes
`timer_create_info_table`		systimer and memory mapped timer nodes
`wd_create_info_table`		watchdog nodes
`pcie_create_info_table`		pcie node
`iovirt_create_info_table`		smmu node
`peripheral_create_info_table`		usb, uart, and sata node
`memory_create_info_table`	gBS->GetMemoryMap	-

3. Linux application

This chapter provides information on executing tests from the Linux application.

3.1 Linux application arguments

This section provides information on the Linux application arguments.

Run the Linux application with the following set of arguments.

```
shell> bsa [-v <n>] [-l <n>] [--only <n>] [-r <ruleids>] [--fr] [--skip <ruleids>
 [--skip-dp-nic-ms] [-h]
```

Table 3-1: Description of Linux application arguments

Argument	Description
`-h`	Prints help message and exits.
`-l`	Level of compliance to be tested for. The default value is 1.
`--only <n>`	Runs only the tests for level n.
`-r <RuleIDs>`	Runs only the specified rules. You can provide a comma-separated list of Rule IDs (for example: `B_PE_01`, `B_GIC_01`). **Note:** Rule ID string passed must match the BSA specification.
`-fr`	Runs rules up to the Future Requirements (FR) level.
`--skip <RuleIDs>`	Skips the specified rules. Provide a comma-separated list of Rule IDs, using the same format as the `-r` option. Example: `-skip B_PE_01,B_GIC_02`
`--skip-dp-nic-ms`	Skips PCIe tests for DisplayPort, network, and mass storage devices.

Example

In the following example, the set of parameters tests for compliance against BSA with print verbosity set to 3, and skips the test for rule ID PCI_IN_01.

```
shell> bsa -v 3 -skip PCI_IN_01
```

Loading the kernel module

Before running the BSA ACS Linux application, load the BSA ACS kernel module using the `insmod` command.

```
shell> insmod bsa_acs.ko
```

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.

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 the 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 for a Beta product, that is a product under development.

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
0102-01	10 December 2025	Non-Confidential	REL v1.2.0
0100-07	28 May 2025	Non-Confidential	REL v1.1.1
0100-06	25 March 2025	Non-Confidential	REL v1.1.0
0100-05	29 March 2024	Non-Confidential	REL v1.0.8
0100-04	28 September 2023	Non-Confidential	REL v1.0.6
0100-03	28 March 2023	Non-Confidential	REL v1.0.4
0100-02	29 October 2022	Non-Confidential	REL v1.0.1
0100-01	6 September 2021	Non-Confidential	REL v1.0
0009-02	26 July 2021	Non-Confidential	Beta release
0005-01	12 May 2021	Non-Confidential	Alpha release

The Change history tables describe the technical changes between released issues of this document in reverse order. Issue numbers match the revision history in Document release information on page 13.

Table 2: Issue 0005-01

Change	Location
First release	-

Table 3: Differences between Issue 0005-01 to Issue 0009-02

Change	Location
Added the abbreviation for SMMU in the list.	See, 1.1 Abbreviations on page 4.
Added a UEFI shell argument with its description.[-dtb <file name>]	See, 2.1 UEFI shell application arguments on page 5.

Table 4: Differences between Issue 0009-02 to Issue 0100-01

Change	Location
Added sata to the PAL APIs, UEFI interfaces, and DT nodes consumed table.	See, 2.2 UEFI shell implementation of PAL APIs on page 7.

Table 5: Differences between Issue 0100-01 to Issue 0100-02

Change	Location
Added abbreviations for HVC and PSCI in the list.	See, 1.1 Abbreviations on page 4.
Added `[-dtb [-t <test id>]`, `[-m <module id>]`, `[-sbsa]`, `[-timeout <wakeup test timeout multiple>]`, `[-p2p]`, `[-cache]` to UEFI shell application arguments	See, 2.1 UEFI shell application arguments on page 5.
Added one more example for UEFI shell application usage.	See, 2.1 UEFI shell application arguments on page 5.

Table 6: Differences between Issue 0100-02 to Issue 0100-03

Change	Location
Added the parameter bsa.efi to UEFI shell arguments.	See, 2.1 UEFI shell application arguments on page 5.
Added PAL APIs.	See, 2.2 UEFI shell implementation of PAL APIs on page 7.

Table 7: Differences between Issue 0100-03 to Issue 0100-04

Change	Location
Updated the UEFI shell application set of arguments and their descriptions.	See, 2.1 UEFI shell application arguments on page 5.

Table 8: Differences between Issue 0100-04 to Issue 0100-05

Change	Location
No technical changes.	-

Table 9: Differences between Issue 0100-05 to Issue 0100-06

Change	Location
Added UEFI shell argument with its description [-l <n>] [-only] [-fr] .	See, 2.1 UEFI shell application arguments on page 5.

Change	Location
Added Linux application argument with its description [-l <n>] [-only] [-fr] .	See, 3.1 Linux application arguments on page 10.

Table 10: Differences between Issue 0100-06 to Issue 0100-07

Change	Location
Added the UEFI shell application arguments.	See, 2.1 UEFI shell application arguments on page 5.

Table 11: Differences between Issue 0100-07 to Issue 0102-01

Change	Location
Updated the UEFI shell application arguments section.	See, 2.1 UEFI shell application arguments on page 5.
Updated the Linux application arguments section.	See, 3.1 Linux application arguments on page 10.
Removed the test ids section.	-

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
italic	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 underline`	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: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
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 of harming yourself.

This information is important and needs your attention.

This information might help you perform a task in an easier, better, or faster way.

This information reminds you of something important relating to the current content.

Useful resources

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

Arm documents are available on developer.arm.com/documentation.

Confidential documents are only available to licensees, when logged in. Each document link in the following tables provides direct access to the online version of the document.

Arm product resources	Document ID	Confidentiality
GICv3 and GICv4 Software Overview	DAI0492	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
Arm® Architecture Reference Manual for A-profile architecture	DDI0487	Non-Confidential
Arm® Generic Interrupt Controller Architecture Specification for GIC architecture version 3.0 and version 4.0	IHI0069	Non-Confidential
Arm® Base System Architecture 1.0	DEN0094C	Non-Confidential