Arm® Base Boot Requirements Architecture Compliance

Revision: r0p9

User Guide

arm

102505_0009_02_en

Arm® Base Boot Requirements Architecture Compliance

User Guide

Release Information

Document History

Issue	Date	Confidentiality	Change
0008-01	18 May 2021	Non-Confidential	Beta release
0009-02	26 July 2021	Non-Confidential	Beta release

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

Web Address

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.

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*.

Contents

Arm® Base Boot Requirements Architecture Compliance User Guide

Preface

This preface introduces the *Arm® Base Boot Requirements Architecture Compliance User Guide*.

It contains the following:

About this book

This book is the user guide for Arm® BBR architecture compliance.

Using this book

This book is organized into the following chapters:

Chapter 1 Introduction to BBR

This chapter provides information on BBR.

Chapter 2 UEFI SCT

This chapter provides information on UEFI SCT.

Chapter 3 Firmware Test Suite

This chapter provides information on FWTS, steps to run, and to build FWTS.

Chapter 4 Building UEFI SCT and FWTS

This chapter provides information on building UEFI SCT and FWTS tests.

Appendix A Revisions

This appendix describes the technical changes between released issues of this book.

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.

Typographic conventions

italic

Introduces special terminology, denotes cross-references, and citations.

bold

Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.

`monospace`

Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.

`monospace`

Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

`monospace italic`

Denotes arguments to monospace text where the argument is to be replaced by a specific value.

`monospace bold`

Denotes language keywords when used outside example code.

`<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

Used in body text for a few terms that have specific technical meanings, that are defined in the *Arm® Glossary*. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

Additional reading

This book contains information that is specific to this product. See the following documents for other relevant information.

Arm publications

- *Arm® Architecture Reference Manual ARMv8, for Armv8-A architecture profile* (ARM DDI 0487G.a (ID011921))
- *Arm® Generic Interrupt Controller Architecture Specification for GIC architecture version 3.0 and version 4.0* (ARM IHI 0069D ID072617)
- *GICv3 and GICv4 Software Overview* (DAI 0492)
- *Arm® Base System Architecture 1.0 Platform Design Document* (DEN0094A)
- *Arm® Base Boot Requirements* (DEN0044F)

Other publications

None.

Feedback

Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give:

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on content

If you have comments on content then send an e-mail to *support-systemready-acs@arm.com*. Give:

- The title *Arm Base Boot Requirements Architecture Compliance User Guide*.
- The number 102505_0009_02_en.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

Note

Arm tests the PDF only in Adobe Acrobat and Acrobat Reader, and cannot guarantee the quality of the represented document when used with any other PDF reader.

Other information

- *Arm® Developer*.
- *Arm® Documentation*.
- *Technical Support*.
- *Arm® Glossary*.

Chapter 1
Introduction to BBR

This chapter provides information on BBR.

It contains the following sections:

1.1 Abbreviations

This section lists the abbreviations used in this document.

Table 1-1 Abbreviations and expansion

Abbreviation	Expansion
ACPI	Advanced Configuration and Power Interface
BBR	Base Boot Requirements
BSA	Base System Architecture
DT	Device Tree
EBBR	Embedded Base Boot Requirements
FWTS	Firmware Test Suite
PSCI	Power State Coordination Interface
SCT	Self-Certification Tests
SBBR	Server Base Boot Requirements
SMBIOS	System Management BIOS
SMCCC	SMC Calling Convention
UEFI	Unified Extensible Firmware Interface

1.2 About the BBR

This section provides information on Base Boot Requirements (BBR) and the runtime executable environments.

BBR for boot and runtime services are based on Arm 64-bit architecture, that system software, for example operating systems and hypervisors, can rely on.

BBR identifies the Arm and industry standard firmware interfaces applicable to the Arm 64-bit architecture. They include the PSCI, SMCCC, UEFI, ACPI, SMBIOS, and DT interfaces. In addition, various recipes are created to accommodate the various operating systems and hypervisors.

The BBR test suites check for compliance against the Server Base Boot Requirements (SBBR) or Embedded Base Boot Requirements (EBBR) specifications. These tests are delivered through two runtime executable environments:

- UEFI SCT
- FWTS

For more information on BBR, see the *BBR specification*.

Chapter 2
UEFI SCT

This chapter provides information on UEFI SCT.

It contains the following section:

2.1 Introduction to UEFI SCT

Unified Extensible Firmware Interface (UEFI) Self-certification Test (SCT) examines the UEFI implementation requirements defined by the BBR recipes that are SBBR or EBBR.

For more information on SCT, see *SCT*.

This section contains the following subsection:

2.1.1 Running UEFI SCT

This section provides information on steps to run UEFI SCTs.

The UEFI SCTs are built as part of the test suite. Running UEFI SCTs is now automated which can be skipped by pressing any key when the corresponding prompt appears from UEFI shell.

```
Shell > Press any key to stop the UEFI SCT running
```

Follow these steps to run the SCT manually:

- ```
  Shell > FS(X):>cd EFI\BOOT\bbr\SCT
  ```
- To run SBBR or EBBR tests:

  ```
  FS(X):EFI\BOOT\bbr\SCT>SCT -s <ebbr.seq/sbbr.seq>
  ```
- To run all the tests:

  ```
  FS(X):EFI\BOOT\bbr\SCT>SCT -a -v
  ```

Note

The correct system time must be set before starting SCT.

You can also select and run tests individually. For more information on running the tests, see *SCT User Guide*.

For more information on steps to build SCT for SBBR or EBBR recipes, see *Chapter 4 Building UEFI SCT and FWTS* on page 4-15.
  ```

Chapter 3
Firmware Test Suite

This chapter provides information on FWTS, steps to run, and to build FWTS.

It contains the following section:

3.1 Introduction to FWTS

Firmware Test Suite (FWTS) is a package that is hosted by Canonical. It provides tests for Advanced Configuration and Power Interface (ACPI), System Management BIOS (SMBIOS), and UEFI. Several SBBR requirements are tested through FWTS.

For more information on FWTS, see *FWTS*.

3.1.1 Running FWTS

To run FWTS manually, the built image must be first copied to the bin directory Linux file system.

Follow these steps to run FWTS on the Linux prompt:

- For SBBR, use the following command:

```
#/bin/fwts -r stdout -q --sbbr
```

- For EBBR, use the following command:

```
#/bin/fwts uefirtmisc uefirttime uefirtvariable securebootcert uefirtauthvar uefivarinfo
esrt uefibootpath
```

Chapter 4
Building UEFI SCT and FWTS

This chapter provides information on building UEFI SCT and FWTS tests.

It contains the following section:

4.1 Building UEFI SCT and FWTS

This section describes how to build UEFI SCT and FWTS for EBBR and SBBR recipes.

The SystemReady ACS live image contains all the BBR ACS tests. However, for debug and verification of bug-fixes, the following steps enable the user to build UEFI SCT and FWTS images independently.

Follow these steps to build UEFI SCT and FWTS:

1. Clone BBR repository:

```
git clone https://github.com/ARM-software/bbr-acs
```

2. Get the source code:

```
cd bbr-acs/<ebbr/sbbr>/scripts
./build-scripts/get_<ebbr/sbbr>_source.sh
```

3. Build BBR ACS from the scripts directory run:

```
./build-scripts/build_<ebbr/sbbr>.sh
```

   Patches are applied by the script to the build files to create EBBR or SBBR image.

Note

- The generated image of UEFI SCT can be found in `bbr-acs/<ebbr/sbbr>/scripts/edk2-test/uefi-sct/<ARCH>_SCT` which is the `AARCH64_SCT`.
- The generated image of FWTS can be found in `bbr-acs/<ebbr/sbbr>/scripts/fwts/fwts_output`.
- In the UEFI application, *`CapsuleApp.efi`* can be found at the location `bbr-acs/<ebbr/sbbr>/scripts/edk2/Build/MdeModule/DEBUG_GCC5/AARCH64`.

Appendix A
Revisions

This appendix describes the technical changes between released issues of this book.

It contains the following section:

- *A.1 Revisions* on page Appx-A-18.

A.1 Revisions

This section consists of all the technical changes between different versions of this document.

Table A-1 Issue 0008-01

Change	Location
First release	-

Table A-2 Issue 0008-01 to Issue 0009-02

Change	Location
Added a note providing location for CapsuleApp.efi.	See, *4.1 Building UEFI SCT and FWTS* on page 4-16.