**# Title:**

Introduce UEFI Conformance Profiles

**# Status:**

Draft

**# Document:**

UEFI Specification 2.9 (Future Errata)

**# License:**

SPDX-License-Identifier: CC-BY-4.0

**# Submitter:**

* Samer El-Haj-Mahmoud, ARM
* TianoCore Community (<https://www.tianocore.org>)

**# Summary of the change**

This proposal updates the UEFI specification to allow publishing UEFI Conformance profiles, allowing implementations to advertise to the OS if they follow a reduced-model of UEFI implementation, which conforms to an external specification defining a subset of the UEFI specification requirements.

**# Benefits of the change**

This proposal updates the UEFI specification to introduce the concept of “Conformance Profiles”, allowing publishing UEFI Conformance profiles, allowing implementations to advertise to the OS if they follow a reduced-model of UEFI implementation. One motivation for such minimized firmware/OS interface is to allow various firmware architecture, such as U-Boot, linuxboot, to implement the UEFI ABI.

**# Impact of the change**

* **Platform FW:**
  + Generally: no impact to existing mainstream UEFI FW implementations (such as those based on TianoCore / EDK2 and derived commercial FW)
  + FW implementations with deviations from UEFI conformance (as defined by external specifications) may choose to publish their profiles to indicate so. A good example is Boot based FW, which may want to advertise the “EBBR v 2.0 Profile”
* **Operating Systems:**
  + OSes may need to be updated to look at known profiles to get insight on the platform FW capabilities without deep probing of interfaces. For instance, Linux may choose to do this to identify UBoot based UEFI FW that is EBBR compliant. This is optional, and may be left until use-cases come up.
* **Compliance tests**
  + Test suites like FWTS and SCT should be updated to understand the profiles, and use in judging the FW compliance to those specifications, as needed

**# Detailed description of the change [normative updates]**

* Insertions highlighted
* Removals in ~~red~~

4.6 EFI Configuration Table & Properties Table

…

**EFI\_MEMORY\_ATTRIBUTES\_TABLE**

**…**

**EFI\_CONFORMANCE\_PROFILE\_TABLE**

**Summary**

This table allows the platform to advertise its UEFI specification conformance in the form of pre-defined profiles. Each profile is identified by a GUID, with known profiles listed in the Description section below.

**Prototype**

#define EFI\_CONFORMANCE\_PROFILES\_TABLE\_GUID \

{ 0x36122546, 0xf7e7, 0x4c8f, \

{ 0xbd, 0x9b, 0xeb, 0x85, 0x25, 0xb5, 0x0c, 0x0b }}

typedef struct {

UINT16 Version;

UINT16 NumberOfProfiles;

//EFI\_GUID ConformanceProfiles [];

} EFI\_CONFORMANCE\_PROFILES\_TABLE;

*Version* Version of the table, must be 0x1

#define EFI\_CONFORMANCE\_PROFILES\_TABLE\_VERSION 0x1

*NumberOfProfiles* The number of conformance profiles GUIDs present in *ConformanceProfiles*.

*ConformanceProfiles* An array of conformance profile GUIDs that are supported by this system.

The address reported in the EFI configuration table entry of this type will be referenced as physical and will not be fixed up when transition from preboot to runtime phase.

**Description**

The following list shows the GUIDs of known conformance profiles. This list is not exhaustive and does not show GUIDs for all possible profiles. Additional profiles can be defined and published in other specifications.

#define EFI\_CONFORMANCE\_PROFILES\_TABLE\_EBBR\_2\_0\_GUID \

{ 0xcce33c35, 0x74ac, 0x4087, \  
{ 0xbc, 0xe7, 0x8b, 0x29, 0xb0, 0x2e, 0xeb, 0x27 }}

Conformance profile defined by the EBBR 2.0.x specification. For more information, see “Links to UEFI-Related Documents” (<http://uefi.org/uefi>) under the heading “EBBR Specification”

**# Additional Instructions**

Add link to <http://uefi.org/uefi>

|  |  |
| --- | --- |
| EBBR Specificaiton | <https://github.com/arm-software/ebbr> |