UEFI Support and Requirements for Windows Operating Systems

April 2, 2013

Abstract

This paper discusses support requirements and guidelines to help improve the end-user experience for Universal Extensible Firmware Interface (UEFI) for the Windows® operating systems.

This document does not replace any existing firmware requirements (such as ACPI requirements).

**Applies to:** Windows 8
 Windows Server 2012
 Windows 7
 Windows Server® 2008 R2
 Windows Vista® SP1 and later
 Windows Server 2008
 Windows Server 2003

References and resources discussed here are listed at the end of this paper.

For the latest information, see:
 <http://www.microsoft.com/whdc/system/platform/firmware/uefireg.mspx>

Disclaimer: This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, Windows, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Change |  |  |  |
| April 2, 2013 | Updated for Windows 8 and Windows Server 2012. |
| April 24, 2009 | Originally targeted for Windows Server 2008, this document now includes UEFI guidance for all supported Windows operating systems. |
| December 11, 2006 | First publication |

Contents

[Introduction 3](#_Toc228256476)

[Firmware Revision Support 3](#_Toc228256477)

[UEFI Support and Requirements 3](#_Toc228256478)

[Installation Requirements 3](#_Toc228256479)

[Installation Media Format 3](#_Toc228256480)

[Hard Disk Partition 5](#_Toc228256481)

[Boot Time Requirements 5](#_Toc228256482)

[Display at Boot Time 5](#_Toc228256483)

[Input at Boot Time 6](#_Toc228256484)

[Network Boot 6](#_Toc228256485)

[Disk Boot 6](#_Toc228256486)

[Other Firmware Boot Requirements 6](#_Toc228256487)

[Runtime Requirements 6](#_Toc228256488)

[Hibernation State (S4) Transition Requirements 6](#_Toc228256489)

[System Memory Requirements 6](#_Toc228256490)

[Firmware Memory Requirements 7](#_Toc228256491)

[Future Requirements to Enable UEFI Platforms without CSM 7](#_Toc228256492)

[Resources 7](#_Toc228256493)

# Introduction

This paper discusses support requirements and guidelines to help improve the end-user experience for Universal Extensible Firmware Interface (UEFI) for the Windows® operating systems.

This document does not replace any existing firmware requirements (such as ACPI requirements).

# Firmware Revision Support

Windows operating systems support firmware revisions that are based on the Unified Extensible Firmware Interface (UEFI) Version 2.0 or later specification on the following architectures:

* 64-bit PCs.
* ARM and 32-bit PCs (Windows 8 only).

Windows supports a subset of the functionality that is defined in the UEFI 2.0 specification. Windows implementations do not explicitly check against higher revisions of the firmware. The operating system supports higher revisions of the firmware if they contain the necessary support for Windows Server 2008 and later editions.

**Windows Server® 2008 R2, Windows Server 2008, and Windows Server 2003**:

* **Intel Itanium PCs** are supported.
Windows also supports firmware revisions that are based on the Extensible Firmware Interface (EFI) Version 1.10 specification on Intel Itanium platforms.
* **Class 3 systems are not supported**, because these operating systems assume the presence of legacy BIOS INT10 support in the firmware, which is not available in a Class-3 UEFI implementation.
* **Class 2 systems must be run in legacy BIOS-compatibility mode** by using a Compatibility Support Module (CSM), so they can use the legacy BIOS INT10 features.

# UEFI Support and Requirements

This section outlines UEFI support and requirements for the following:

* Installation
* Boot time
* Runtime
* Hibernation power state (S4) transitions
* UEFI platforms without a Compatibility Support Module (CSM)

## Installation Requirements

To install Windows, Windows Setup must be run in UEFI mode, rather than BIOS-compatibility mode.

You may need to manually select the UEFI boot files to make sure you're booted into the correct mode.

### Detect whether Windows PE is booted in BIOS or UEFI mode

If you're using Windows PE to run Windows Setup, you can run a check to make sure you're booted into the correct mode.

Check the registry to see if the PC is booted to UEFI or BIOS mode.

wpeutil /UpdateBootInfo

reg query HKLM\System\CurrentControlSet\Control /v PEFirmwareType

This command returns 0x1 if the PC is booted into BIOS mode, or 0x2 if the PC is booted in UEFI mode.

Here’s a sample script that queries the registry value:

wpeutil /UpdateBootInfo

for /f "tokens=2\* delims= " %%A in ('reg query HKLM\System\CurrentControlSet\Control /v PEFirmwareType') DO SET Firmware=%%B

:: Note: delims is a TAB followed by a space.

if %Firmware%==0x1 echo The PC is booted in BIOS mode.

if %Firmware%==0x2 echo The PC is booted in UEFI mode.

### Installation Media Format

This section describes the installation media formats and default boot behavior for UEFI platforms.

#### Media Format for x64, x86, or ARM-based PCs

The following guidelines apply for installation media on x64 platforms:

* Windows installation media supports boot on both BIOS and UEFI platforms by taking advantage of multi-entry El Torito boot catalog support.
* The default El Torito boot entry is a BIOS entry that includes the 80x86 Platform ID, which is defined as “0x00” in hexadecimal.
* The second El Torito boot entry is an EFI entry that includes the Platform ID as “0xEF” in hexadecimal. The entry references a FAT partition that contains the bootable EFI application at \EFI\BOOT\BOOTX64.EFI.

#### Media Format for IA64 Platform

The following guideline applies for installation media on IA64 platforms:

* The media contains a single El Torito boot entry that defines the Platform ID as “0xEF” in hexadecimal. The entry references a FAT partition that contains the bootable EFI application at \EFI\BOOT\BOOTIA64.EFI.

#### Default Boot Behavior

Firmware vendors must ensure that the following conditions exist:

* The BIOS ignores boot entries that do not have the 80x86 Platform ID, which is defined as “0x00” in hexadecimal. Failure to ignore other boot entries results in the display of a confusing boot menu to the end user.
* The BIOS boots based on the BIOS entry without prompt.
* The UEFI boot manager ignores boot entries that do not have the “0xEF” Platform ID.
* The UEFI boot manager boots based on the EFI entry without prompt.

To support the ability to boot from DVD media, the Windows installation DVD contains many El Torito boot entries that enable boot from either BIOS or UEFI. The default El Torito boot entry is for BIOS.

Windows 8, Windows 7, Server 2012, and Server 2008 R2 include support for the “Non-removable Media Boot Behavior” section that was added to the UEFI 2.3 specification. These versions of Windows copy the Windows boot application from \efi\microsoft \boot\bootmgfw.efi to \efi\boot\boot{*arch*}.efi on the EFI system partition during initial installation and when updates are required for bootmgfw.efi. This copy enables a default boot option for Windows if a nonvolatile RAM (NVRAM) boot entry is not available, such as when a hard disk is moved from one platform to another.

#### Other Media Information

The following additional guidelines apply for boot media:

* Windows supports both CD and DVD boot from the Universal Disk Format (UDF) file system. Windows installation media also uses El Torito and is built by using the UDF bridge format to support both ISO 9660 and UDF version 1.02 file systems.
* The Windows OEM Preinstallation Kit (OPK) and Windows Automated Installation Kit (WAIK) include an updated version of Oscdimg.exe that supports the creation of a multi-entry El Torito boot catalog.

### Hard Disk Partition

Windows requires the operating system partition to reside on a GUID partition table (GPT) partitioned disk. Master boot record (MBR) partitioned disks can be used as data disks. As defined in the UEFI 2.0 specification, a UEFI platform requires a dedicated system partition. Windows requires this dedicated system partition, which is referred to as the EFI system partition (ESP).

## Boot Time Requirements

This section describes UEFI and firmware requirements for the following at boot time:

* Display
* Input
* Network
* Disk
* Other requirements

### Display at Boot Time

For a platform that has a console device, the UEFI 2.0 specification requires the firmware to implement the Simple Text Output Protocol. Optionally, the firmware can also support a graphical protocol. UEFI 2.0 defines the Graphic Output Protocol (GOP), and EFI 1.1 defines the Universal Graphics Adapter (UGA) Protocol. Windows supports all three protocols, but the user experience with each protocol is different. For the best experience, if the firmware implements a graphical protocol, Windows recommends and prefers the GOP.

Windows requires a graphical protocol to render glyphs for non-English message resources. To do so, the firmware must support the following:

* A graphical protocol—either GOP or UGA.
* Either 1024x768 display resolution with 32-bit pixel color or 800x600 display resolution with 24-bit pixel color.

If the firmware does not support any of these graphics modes, Windows still functions, but all boot display reverts to text mode and English.

Windows 8, Windows 7, Server 2012, and Server 2008 R2 require GOP to display a high-resolution, animated image during boot. If GOP is not available, Windows uses the video graphics array (VGA) standard to display a lower resolution image and a simple progress indicator. For an optimal boot experience with these versions of Windows, sealed platforms without expansion card slots can safely boot with graphics mode enabled and eliminate transitions to text mode.

Whenever the firmware boot manager hands off execution to a Windows EFI application, platform firmware and the firmware boot manager must not use the frame buffer for any purpose.

### Input at Boot Time

For a platform that has a console device, the UEFI 2.0 specification requires the firmware to implement the Simple Input Protocol. Windows supports this protocol for local keyboard input during boot.

### Network Boot

Windows implements support for the EFI Preboot eXecution Environment (PXE) Base Code Protocol. Windows uses this protocol to boot over the network and support Windows Deployment Services (WDS).

### Disk Boot

Windows requires Block I/O Protocol and Device Path Protocol support for the disk that contains the EFI system partition and the Windows OS partition.

### Other Firmware Boot Requirements

To ensure proper operation, Windows requires EFI firmware to comply with its indicated specification version. EFI firmware must fully implement the appropriate version of the EFI System Table, EFI Boot Services, and EFI Runtime Services. Other specific required protocols and specifications include the following:

* **Trusted Computing Group (TCG) EFI Specifications**. All UEFI platforms that have a Trusted Platform Module (TPM) must implement the TCG EFI Platform and Protocol specifications.
* **EFI\_PCI\_ROOT\_BRIDGE\_IO\_PROTOCOL**. Windows uses this protocol if Windows Boot Configuration Data (BCD) specifies IEEE 1394 boot debugging.

## Runtime Requirements

Windows minimizes its use of UEFI services during operating system runtime and, wherever possible, relies on runtime firmware such as ACPI and Windows drivers. Windows uses the following UEFI Runtime Services to manage NVRAM boot entries and hardware error records after ExitBootServices() is called.

* GetVariable
* GetNextVariableName
* SetVariable
* QueryVariableInfo

## Hibernation State (S4) Transition Requirements

This section describes requirements for system and firmware memory that are related to transitions to the hibernation power state (S4).

### System Memory Requirements

Platform firmware must ensure that operating system physical memory is consistent across S4 sleep state transitions, in both size and location.

Operating system physical memory is defined according to the ACPI 3.0 specification as any memory that is described by the firmware system address map interface with a memory type other than *AddressRangeReserved* [2], *AddressRangeUnusable* [5], or *Undefined* [any value greater than 5].

### Firmware Memory Requirements

On a UEFI platform, firmware runtime memory must be consistent across S4 sleep state transitions, in both size and location. Runtime memory is defined according to the UEFI specification as any memory that is described by the GetMemoryMap() boot service, with the attribute *EFI\_MEMORY\_RUNTIME*.

## Requirements to Enable UEFI Platforms without CSM

First-generation 64-bit UEFI platforms typically contain some form of limited BIOS emulation such as a CSM to preserve the ability to run 32-bit operating systems and operating systems that do not support UEFI. Existing Windows dependencies on INT 10 video BIOS functions also require a CSM.

To reduce the need for a CSM and improve boot times in the future, we are collaborating with the industry to eliminate this dependency and encourage changes to system firmware.

We have already identified the following future firmware requirements:

* **GOP**. Windows uses the GOP to obtain a frame buffer pointer at boot time for use during operating system runtime. GOP support is essential to replace VGA support and avoid the requirement for a CSM in future versions of Windows.
* **EFI Capsule Services**. Future versions of Windows can use the EFI UpdateCapsule() service to persist information across a system restart and pass that information to the firmware. This would potentially let the system report and/or respond to certain error conditions if the boot device or operating system were damaged or otherwise unavailable.

### Firmware recommendations

Note to firmware manufacturers: We recommend that when Secure Boot is disabled, then the firmware should trigger the following actions, to provide a better support experience for previous versions of Windows:

* Enable the CSM for VGA support, though not BIOS mode emulation.
* Enable messages during the POST process to show which keys open the boot menus.

# Resources

ACPI 3.0 Specification

<http://www.acpi.info/spec30.htm>

EFI Specification Version 1.10

<http://www.intel.com/technology/efi/main_specification.htm>

UEFI Specification Version 2.0 and later

<http://www.uefi.org/specs/>