How to Identify the Windows Version in ACPI by Using \_OSI

April 17, 2017

Abstract

This paper describes how to use the \_OSI method in Advanced Configuration and Power Interface (ACPI) Source Language (ASL) to identify the host operating system. By using this method, ASL writers can create firmware that supports future operating systems versions and enables the operating system to change behavior based on the requested interface levels.

This information applies to the following operating systems:

 Windows 10, version 1703

 Windows 10, version 1607
 Windows Server Technical Preview
 Windows 10
 Windows Server 2012 R2
 Windows 8.1
 Windows Server 2012
 Windows 8
 Windows Server 2008 R2
 Windows 7
 Windows Server 2008
 Windows Vista
 Windows Server 2003
 Windows XP

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

The current version of this paper is maintained on the Web at:
 [How to Identify the Windows Version in ACPI by Using \_OSI](http://msdn.microsoft.com/library/windows/hardware/gg463275)

**Disclaimer**: This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet website references, may change without notice. Some information relates to pre-released product which may be substantially modified before it’s commercially released. Microsoft makes no warranties, express or implied, with respect to the information provided here. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2015 Microsoft. All rights reserved.



Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Change |  |  |  |
| April 17, 2017 | Updated document for Windows 10, version 1607 and Windows 10, version 1703 |
| March 17, 2015 | Updated document for Windows 10 and Windows Server Technical Preview |
| August 5, 2013 | Updated document for Windows 8.1 and Windows Server 2012 R2. |
| July 13, 2012 | Updated document for Windows 8 and Windows Server 2012. |
| October 1, 2012 | Updated document for Windows 7 and Windows Server 2008 R2. |
| January 15, 2003 | First publication |

Contents

[The \_OSI Method 3](#_Toc480276369)

[\_OSI Defined 3](#_Toc480276370)

[\_OSI Argument Details for Windows 3](#_Toc480276371)

[Implementation Note 4](#_Toc480276372)

[References and Resources 4](#_Toc480276373)

# The \_OSI Method

All recent versions of the Windows operating system support components of the *Advanced Configuration and Power Interface (ACPI) Specification*. The ACPI specification defines an interpreted language, ACPI Source Language (ASL), to enable the operating system to execute firmware-provided control methods for power management and configuration. To improve the ability for ASL writers to identify the host operating system version, ASL provides the Operating System Interface Level (\_OSI).

By using the \_OSI method, ASL writers can easily determine the version of the ACPI interfaces that the host operating system supports. This versioning method provides a solution for creating firmware that can support future operating systems and enable the operating system to change behavior based on the requested interface levels.

# \_OSI Defined

The \_OSI method has one argument and one return value. The argument is a string that is defined by and for each operating system. The return value is 0x00000000 if the interface is not supported or 0xFFFFFFFF if the interface is supported.

Recent versions of the ACPI specification have extended the use cases of the \_OSI method beyond host operating system version identification. However, Windows supports \_OSI only for the use of identifying the host version of Windows that is running on the system.

The \_OSI method is defined as follows:

\\_OSI - Operating System Interfaces

#### Argument

A string defined by and for each operating system. For example:

“Windows 2013” for Windows 8.1 and Windows Server® 2012 R2
“Windows 2012” for Windows 8 and Windows Server® 2012
“Windows 2009” for Windows 7 and Windows Server® 2008 R2
“Windows 2001” for Windows XP
“Windows 2001.1” for Windows Server 2003

#### Return Value

0x00000000 if the operating system does not support the version in the argument.
0xFFFFFFFF if the operating system does support the version in the argument.

## \_OSI Argument Details for Windows

Table 1 lists the versions of Windows that ASL can identify by using the corresponding \_OSI string.

Windows operating systems return 0xFFFFFFFF if the argument to the \_OSI method specifies an earlier version of Windows. For example, Windows 7 returns 0xFFFFFFFF for both “Windows 2009” (Windows 7) and “Windows 2006” (Windows Vista®).

Table 1. \_OSI Strings for Windows Operating Systems

|  |  |
| --- | --- |
| Windows operating system version | \_OSI string |
| Windows 10, version 1703 | “Windows 2017” |
| Windows 10, version 1607 | “Windows 2016” |
| Windows 10 | “Windows 2015” |
| Windows Server Technical Preview | “Windows 2015” |
| Windows 8.1 | “Windows 2013” |
| Windows Server 2012 R2 | “Windows 2013” |
| Windows 8 | “Windows 2012” |
| Windows Server 2012 | “Windows 2012” |
| Windows 7 | “Windows 2009” |
| Windows Server 2008 R2 | “Windows 2009” |
| Windows Vista  | “Windows 2006” |
| Windows Vista SP1 | “Windows 2006 SP1” |
| Windows Server 2008 | “Windows 2006.1” |
| Windows XP | “Windows 2001” |
| Windows XP SP2 | “Windows 2001 SP2” |
| Windows Server 2003 | “Windows 2001.1” |
| Windows Server 2003 SP1 | “Windows 2001.1 SP1” |

## Implementation Note

Place the routine that identifies the operating system in an \_INI method under the \\_SB scope so that \_OSI can run as early as possible. This placement is important because the operating system makes features available based on the string argument to the \_OSI method.

# References and Resources

Advanced Configuration and Power Interface Specification

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