



SMBIOS Support in Windows

WinHEC 2005 Update - April 20, 2005

Abstract

This paper provides information about System Management BIOS (SMBIOS) support for the Microsoft® Windows® family of operating systems. It provides guidelines for system manufacturers and software developers to publish and interact with SMBIOS data on Windows systems.

This information applies for the following operating systems:

- Microsoft Windows Vista
- Microsoft Windows Server codename “Longhorn”
- Microsoft Windows Server™ 2003 with Service Pack 1
- Microsoft Windows XP with Service Pack 2

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

Contents

Introduction	3
Reading SMBIOS Table Information from Applications	3
Using Windows Management Instrumentation to Read SMBIOS Data	3
System APIs for Reading SMBIOS Data	3
Driver Access to SMBIOS Data by Using WMI	4
Microsoft System Management BIOS Driver	4
Data Collection	4
System Identification	6
SMBIOS “Designed for Windows” Logo Requirements	8
Next Steps	8
Additional Resources	8

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.

© 2005 Microsoft Corporation. All rights reserved.

Microsoft, Win32, Windows, and Windows Server 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.

Introduction

The Microsoft Windows family of operating systems provides robust support for system management BIOS (SMBIOS) data. Management programs often use SMBIOS data to discover information about the hardware platform, such as the system manufacturer or the system BIOS version.

Windows relieves programs of the need to scan physical memory for SMBIOS table data by automatically retrieving SMBIOS data at system startup and then making it available to programs. SMBIOS data is available through Windows Management Instrumentation (WMI) classes and through the Microsoft Win32® application programming interface (API).

Windows XP SP2 and Windows Server 2003 SP1 include a new driver, the Microsoft SMBIOS driver, which offers support for additional data collection capabilities and accurate system identification.

Future versions of Windows will depend on accurate system identification for new scenarios. This paper explains the requirements for accurate system identification using SMBIOS, information that will be important to system manufacturers. For software developers, this paper explains how to access SMBIOS data in Windows.

Reading SMBIOS Table Information from Applications

The following methods can be used to read SMBIOS table data from an application.

Using Windows Management Instrumentation to Read SMBIOS Data

WMI is the preferred method for reading SMBIOS information in Windows. WMI supports reading individual SMBIOS properties, such as the BIOS version or the system manufacturer, from Win32_xxx classes in the Root\Cimv2 namespace. WMI also supports reading the entire contents of SMBIOS data in a single buffer by using the MSSMBios_RawSMBiosTables class inside the root\wmi namespace. The SMBiosData property returns a buffer that contains the entire SMBIOS data table, except the SMBIOS structure table entry point as defined in SMBIOS Specification, Section 2.1.1, available from the Distributed Management Task Force.

WMI supports a managed COM and a script API to access WMI information. More information about APIs to access WMI information can be found in the WMI SDK.

Additionally, WMI is supported in the Windows XP SP2 and Windows Server 2003 SP1 versions of the Windows Preinstallation Environment (WinPE). WinPE is commonly used in system manufacturing and image deployment scenarios.

System APIs for Reading SMBIOS Data

Beginning with Windows Server 2003 SP1, applications can call two new user-mode functions to retrieve various system firmware tables, including SMBIOS tables. To read SMBIOS and ACPI tables and to read raw firmware data from low physical memory, programs can call **EnumSystemFirmwareTables()** and **GetSystemFirmwareTable()**. Information about these functions is available on the Microsoft Developer Network website.

Driver Access to SMBIOS Data by Using WMI

Kernel-mode device drivers can use WMI to read the raw SMBIOS data tables in the same manner as user-mode applications access the MSSMBios_RawSMBiosTables WMI class.

Kernel-mode device drivers can call the **IoWmiOpenBlock()** and the **IoWmiQueryAllData()** functions to retrieve a buffer that contains the raw contents of the SMBIOS table data. Specifically, this buffer contains the same SMBIOS data as the buffer returned by calling the **GetSystemFirmwareTable()** function with the raw SMBIOS (RSMB) table provider.

The returned buffer has the following data:

BYTE	Used20CallingMethod
BYTE	SMBIOSMajorVersion
BYTE	SMBIOSMinorVersion
BYTE	DmiRevision
DWORD	Length
BYTE[]	SMBIOSTableData

Each value in the buffer return corresponds to the individual properties stored in the MSSMBios_RawSMBiosTables WMI class. The SMBIOSTableData property contains the entire SMBIOS data table, except for the SMBIOS structure table entry point.

You can achieve access to the SMBIOS data in the MSSMBios_RawSMBiosTables class by using the class GUID, {8F680850-A584-11d1-BF38-00A0C9062910}.

Microsoft System Management BIOS Driver

The Microsoft SMBIOS driver (mssmbios.sys) is a root-enumerated driver included with Windows XP SP2 and Windows Server 2003 SP1 and has two primary functions:

- To incorporate SMBIOS, ACPI, and CPU information to the crash dump file when a system crash occurs.
- To retrieve system identification information from SMBIOS and make it available to system components and applications.

There are two primary benefits to the Microsoft SMBIOS driver:

- Adding system firmware information to the crash dump aids in failure diagnosis. A given root cause may be linked to a specific system vendor or to specific hardware attributes, such as the type and size of memory modules. This information is available in SMBIOS.
- Identifying a given hardware platform. Scenarios such as branding, BIOS update, and targeted driver installation require a reliable, stable method for identifying the underlying hardware platform. These scenarios require information about basic attributes of the system, such as the system manufacturer and current BIOS version.

Data Collection

At system startup, the Microsoft SMBIOS driver collects the following information about the system hardware and stores it in memory allocated by the driver:

- **SMBIOS data:** The SMBIOS driver collects the entire contents of the SMBIOS table data except for unique identification information. To prevent the data that

is collected from being able to uniquely identify a system, the driver reads the entire contents of the SMBIOS table data and then zeroes out the following SMBIOS data fields if they are present:

- System Information (Type 1 Table)
 - Serial Number
 - UUID
- Base Board Information (Type 2 Table)
 - Serial Number
 - Asset Tag
- System Enclosure Information (Type 3 Table)
 - Serial Number
 - Asset Tag
- Processor Information (Type 4 Table)
 - Serial Number
 - Asset Tag
- Memory Device Information (Type 17 Table)
 - Serial Number
 - Asset Tag
- Battery Information (Type 22 Table)
 - Serial Number
 - SBDS Serial Number
- Power Supply Information (Type 39 Table)
 - Serial Number
 - Asset Tag

After the driver has removed the contents of these fields, it stores the SMBIOS data in memory until a system crash occurs.

- **ACPI information:** The driver collects all ACPI table data except for the contents of the differentiated system description table (DSDT) and secondary system description table (SSDT), if they are present. The driver also collects the header data of the DSDT and SSDT and then stores this table data in memory until a system crash occurs.
- **CPU information:** The driver includes the entire contents of the CPU information key in the registry (HKLM\Hardware\Description\System\CentralProcessor\0). Some vendor-specific information about the processor, such as the current microcode update revision, is also collected. The driver then stores this data in memory until a system crash occurs.

The driver also stores this information in the registry at HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Mssmbios\Data. Although SMBIOS information is stored in the registry at this location, consumers should continue to use WMI or the GetSystemFirmwareTable() API to retrieve SMBIOS data. There is no guarantee that this information will be stored at this registry subkey for every subsequent release of Windows.

The Microsoft SMBIOS driver registers with the Windows kernel so that the driver is notified whenever a system crash occurs. When this notification occurs, the driver provides the contents of the SMBIOS, ACPI, and CPU data to the kernel and adds this information to the crash dump file.

As described above, the Microsoft SMBIOS driver zeroes out all information that could uniquely identify a system; however, OEMs may include custom information in the OEM strings (Type 11) table or in custom SMBIOS tables. OEMs must therefore ensure that OEM string data and custom table data do not contain information that can be used to uniquely identify the system.

System Identification

The Microsoft SMBIOS driver included with Windows Server 2003 SP1 and later versions of Windows supports system identification, which takes advantage of features in SMBIOS Specification, Version 2.4.

Note: The system identification functionality of the Microsoft SMBIOS driver is not available on Windows XP SP2.

The goals of system identification are to:

- Support new scenarios that depend on correct identification of the underlying system hardware, such as targeted driver installation, BIOS updating, and branding.
- Provide a reliable method for system components and applications to identify the underlying system hardware.
- Ensure that system components and applications can access this data through the registry and discover matches on the data by using a single WMI query language (WQL) query.
- Ensure that components can query for specific pieces of system identification information that are relevant to their scenario. For example, a branding component might need only information about the system manufacturer and the system product name; however, a component responsible for BIOS update might require many additional pieces of system identification information, such as the BIOS major and minor versions or the baseboard version.
- Provide a stable method of system identification. To be useful, system identification information must be derived directly from the BIOS and must be guaranteed by the OEM.

System identification using the Microsoft SMBIOS driver depends on specific features of SMBIOS. The requirements for system identification are:

- System SMBIOS implementation must be Version 2.4 or greater.
- System SMBIOS implementation must have the Enable Targeted Content Distribution bit set in the BIOS Characteristics Extension Byte 2 field of the Type 0 SMBIOS table.
- System SMBIOS implementation must have a single Type 1 (System Information) table of length at least 1Ah bytes (includes SKU Number and Family fields from Version 2.4 of the Type 1 table).
- System SMBIOS implementation must provide non-null values for the following SMBIOS fields:
 - Table 0, offset 04h (BIOS Vendor)
 - Table 0, offset 08h (BIOS Release Date)
 - Table 0, offset 14h (BIOS Major Release Version)

- Table 0, offset 15h (BIOS Minor Release Version)
- Table 1, offset 04h (System Manufacturer)
- Table 1, offset 05h (System Product Name)

If the system SMBIOS implementation meets all of the above requirements, the Microsoft SMBIOS driver exposes the following system identification information in the registry and in WMI:

SMBIOS table type	SMBIOS field name	Field data type
0 (BIOS)	Vendor	string
0 (BIOS)	BIOS Version	string
0 (BIOS)	BIOS Release Date	string
0 (BIOS)	System BIOS Major Release	uint8
0 (BIOS)	System BIOS Minor Release	uint8
0 (BIOS)	Embedded Controller Firmware Major Release	uint8
0 (BIOS)	Embedded Controller Firmware Minor Release	uint8
1 (System)	Manufacturer	String
1 (System)	Product Name	String
1 (System)	Family	String
1 (System)	SKU Number	String
1 (System)	Version	String
2 (Baseboard)	Baseboard Manufacturer	String
2 (Baseboard)	Baseboard Product Name	String
2 (Baseboard)	Baseboard Version	String

The system identification information is retrieved from the SMBIOS data fields and written to the registry at the HKLM\Hardware\Description\BIOS subkey.

The system identification information is also made available through WMI in the MS_SystemInformation class. The MS_SystemInformation WMI class is located in the root\wmi namespace and has the following Managed Object Format (MOF) description:

```
[WMI, dynamic: ToInstance, provider("WMIProv"),
Guid("{98A2B9D7-94DD-496a-847E-67A5557A59F2}"),
Locale(1033)]
class MS_SystemInformation
{
    [key, read] string InstanceName;
    [read] boolean Active;
    [read, WMIDataID(1)] string
BaseBoardManufacturer;
    [read, WMIDataID(2)] string BaseBoardProduct;
    [read, WMIDataID(3)] string BaseBoardVersion;
    [read, WMIDataID(4)] string BIOSReleaseDate;
    [read, WMIDataID(5)] string BIOSVendor;
    [read, WMIDataID(6)] string BIOSVersion;
    [read, WMIDataID(7)] string SystemFamily;
    [read, WMIDataID(8)] string SystemManufacturer;
    [read, WMIDataID(9)] string SystemProductName;
    [read, WMIDataID(10)] string SystemSKU;
    [read, WMIDataID(11)] string SystemVersion;
    [read, WMIDataID(12)] uint8 BiosMajorRelease;
    [read, WMIDataID(13)] uint8 BiosMinorRelease;
    [read, WMIDataID(14)] uint8 ECFirmwareMajorRelease;
    [read, WMIDataID(15)] uint8
ECFirmwareMinorRelease;
```

```
} ;
```

Programs can use WMI or registry functions to access the system identification information. A program can determine if system identification is supported on a given system by the presence of the associated registry values or by an instance of the MS_SystemInformation class.

Although Windows requires non-null values only for some of the system identification information fields, you should provide accurate values for all of the system identification information fields.

For example, system manufacturers should place accurate product SKU information in the SMBIOS SystemSKU field. This information is stable with respect to the system preload image and is added automatically to system crash reports.

Note: Although the Microsoft SMBIOS driver is available on both Windows XP SP2 and Windows Server 2003 SP1, the system identification functionality is available only on Windows Server 2003 SP1 systems. Instances of the MS_SystemInformation class are not created or populated on Windows XP SP2 systems.

SMBIOS “Designed for Windows” Logo Requirements

The “Designed for Windows” Logo Program for Hardware will feature a new requirement concerning system SMBIOS implementation. The SMBIOS implementation on a system meeting the logo requirements must comply with SMBIOS Specification, Version 2.4 or later.

Additionally, the system SMBIOS implementation must follow all conformance requirements as outlined in SMBIOS Specification, Section 3.2, and it must provide accurate values for some of the BIOS, system, and base board information fields.

For specific information about this logo requirement, review the proposed requirements for the logo program.

Next Steps

For Software Developers

Software developers who require read access to the entire set of SMBIOS data should use Windows Management Instrumentation (WMI) or the new firmware table provider APIs for Windows Server 2003 SP1.

For System Manufacturers and BIOS Developers

System manufacturers and BIOS developers should:

- Be aware of the system identification functionality in the Microsoft System Management BIOS driver.
- Implement support for System Management BIOS Specification, Version 2.4, and should set the Enable Targeted Content Distribution bit.
- Be aware that beginning with Microsoft Windows Vista, the “Designed for Windows” Logo Program for Hardware will require support for System Management BIOS Specification, Version 2.4, and for the Enable Targeted Content Distribution bit.

Additional Resources

For more information about WinPE, send e-mail to askwinpe@microsoft.com.

Windows Management Instrumentation (WMI):

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/wmi_start_page.asp

System Management BIOS (SMBIOS) Specification:

<http://www.dmtf.org/standards/smbios>

Windows Preinstallation Environment (WinPE):

<http://www.microsoft.com/licensing/programs/sa/support/winpe.msp>

“Designed for Windows” Logo Program for Hardware:

<http://www.microsoft.com/whdc/winlogo/default.msp>

Windows Quality Online Services:

<http://winqual.microsoft.com/>

WMI Software Developers Kit:

<http://www.microsoft.com/msdownload/platformsdk/sdkupdate/>

Microsoft Developer Network website:

<http://msdn.microsoft.com/>

Proposed Logo Program Requirements for Hardware:

<http://www.microsoft.com/whdc/winlogo/hwrequirements.msp>

System Management BIOS Specification:

<http://www.dmtf.org>