



## PCI-SIG ENGINEERING CHANGE REQUEST

<b>TITLE:</b>	Ignore PCI Boot Configuration _DSM function
<b>DATE:</b>	4/24/2006
<b>AFFECTED DOCUMENT:</b>	PCI Firmware Specification Ver 3.0
<b>SPONSOR:</b>	Tony Pierce, Microsoft

### Part I

#### 1. 1. **Summary of the Functional Changes**

Add a function to the \_DSM Definitions for PCI to provide an indication to an operating system that it can ignore the PCI boot configuration setup by the firmware during system initialization.

Formatted: Bullets and Numbering

#### 2. 2. **Benefits as a Result of the Changes**

Provides a mechanism for the platform to indicate to an operating system that boot configurations can be ignored for a device hierarchy

Allows firmware to thoroughly boot configure devices for best backward compatibility with with current operating systems and provides a smooth transistion to future operating systems

Formatted: Bullets and Numbering

#### 3. 3. **Assessment of the Impact**

High for new operating systems that have the goal of providing flexible resource management

Formatted: Bullets and Numbering

#### 4. 4. **Analysis of the Hardware Implications**

None

Formatted: Bullets and Numbering

#### 5. 5. **Analysis of the Software Implications**

Low, \_DSM is an optional control method

Formatted: Bullets and Numbering

**Part II**

**Detailed Description of the change**

Add Ignore PCI Boot Configuration function to the \_DSM definitions for PCI as function 5 in Table 4-7 on page 56 as follows:

**4.6 \_DSM Definitions for PCI**

\_DSM (Device Specific Method) is defined in the ACPI 3.0 Specification. This object is a control method that enables devices to provide device specific control functions that are consumed by the device driver. Table 5-1 below lists the UUID, revision, and function definitions.

**Table 5-1: \_DSM Definitions for PCI**

UUID	Revision	Function	Description
E5C937D0-3553-4d7a-9117-EA4D19C3434D	1	1	PCI Express Slot Information
	1	2	PCI Express Slot Number
	1	3	Vendor-specific Token ID
	1	4	PCI Bus Capabilities
	<u>1</u>	<u>5</u>	<u>Ignore PCI Boot Configuration</u>

- Formatted: Font color: Black
- Deleted: Table 4-7
- Formatted: Font color: Black, Do not check spelling or grammar
- Formatted: Font color: Black
- Deleted: 4
- Deleted: 7

*Insert the following section after 4.6.4 and before 4.7*

**4.6.5. DSM for Ignoring PCI Boot Configurations**

This section describes how system firmware can indicate to an operating system that it can ignore the boot configurations of PCI hierarchies and devices. The indication is exposed through the \_DSM ACPI method.

Deleted: y

This \_DSM function is optional. If the platform does not provide it, the OS may continue to use the legacy handling regarding the boot configuration.

Note: Function 0 is a generic Query function that is supported by \_DSMs with any UUID and Revision ID. The definition of function 0 is generic to \_DSM and specified in the ACPI Specification, Version 3.0.

Deleted: ¶

**Location:**

This object can be placed under any virtual PCI-to-PCI bridge object representing the PCI Express root port or switch port, PCI Express endpoint, Conventional PCI/PCI-X bridge, or Conventional PCI/PCI-X device. When the object is placed on a bridge, the operating system will assume that the return value applies to the complete hierarchy produced by that bridge.

Note:

## Request Request Request Request Request Request Request Request

The resources for a PCI root bus are described using an CRS method. If an operating system needs to change the resource assignment for a PCI root bus, it needs to examine the PRS methods, select an appropriate resource list for the root bus and set them using an SRS method.

### Arguments:

Arg0: UUID: E5C937D0-3553-4d7a-9117-EA4D19C3434D

Arg1: Revision ID: 1

Arg2: Function Index: 5

Arg3: Empty Package

**Return:**

An integer whose description is as follows:

Formatted: Font: Garamond, 12 pt

Type: Integer

Purpose: Ignore PCI Boot Configuration

Description:

0: No (The operating system shall not ignore the PCI configuration that firmware has done at boot time. However, the operating system is free to configure the devices in this hierarchy that have not been configured by the firmware. There may be a reduced level of hot plug capability support in this hierarchy due to resource constraints. This situation is the same as the legacy situation where this DSM is not provided.)

1: Yes (The operating system may ignore the PCI configuration that the firmware has done at boot time, and reconfigure/rebalance the resources in the hierarchy.)

Formatted: Indent: Left: 0.5"



**IMPLEMENTATION NOTE**

This DSM function provides backwards compatibility on platforms that can run legacy operating systems.

Operating systems for two different architectures (e.g., x86 and x64) can be installed on a platform. The firmware cannot distinguish the operating system in time to change the boot configuration of devices. Say for instance, an x86 operating system in non-PAE mode is installed on a system. The x86 operating system cannot access device resource space above 4GB. So the firmware is required to configure devices at boot time using addresses below 4GB. On the other hand, if an x64 operating system is installed on this system, it can access device resources above the 4GB so it does not want the firmware to constrain the resource assignment below 4GB that the firmware configures at boot time. It is not possible for the firmware to change this by the time it boots the operating system. Ignoring the configurations done by firmware at boot time will allow the operating system to push resource assignment using addresses above 4GB for an x64 operating system while constrain it to addresses below 4GB for an x86 operating system.

Formatted: Font: Garamond, 12 pt