



# **EHCI Removal from 6<sup>th</sup> Generation Intel<sup>®</sup> Core<sup>™</sup> Processor Family Platform Controller Hub (PCH)**

**Technical White Paper**

---

***September 2015***

***Revision 1.0***



You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com](http://intel.com).

Intel technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visit [www.intel.com/design/literature.htm](http://www.intel.com/design/literature.htm)

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries. See Trademarks on [intel.com](http://intel.com) for full list of Intel trademarks.

\*Other names and brands may be claimed as the property of others.

Copyright © 2015, Intel Corporation. All rights reserved.



# Contents

---

1	Introduction .....	5
	1.1 Planned Chipset Changes.....	5
	1.2 Document Scope .....	5
	1.3 Overview .....	5
	1.4 USB Storage Device Support .....	5
	1.5 xHCI-based Windows* 7 Install Flow .....	6
2	xHCI USB Operating System Install Solutions.....	7
	2.1 Adding USB 3.0 Drivers to Windows* 7 Install .....	7
	2.2 Boot from Customized WinPE with USB 3.0 Driver .....	9



## *Revision History*

---

Document Number	Revision Number	Description	Revision Date
333136-001	1.0	Initial release.	September 2015

§ §



# 1 Introduction

---

## 1.1 Planned Chipset Changes

In the 6<sup>th</sup> Generation Intel<sup>®</sup> Core<sup>™</sup> processor family PCH generation, Intel plans to remove both EHCI controllers along with their integrated rate matching hubs (RMH). Intel will continue to provide USB functionality through the xHCI controller for USB 2.0 and USB 3.0 connectivity.

## 1.2 Document Scope

This document discusses the implications of removing EHCI from the 6<sup>th</sup> Generation Intel<sup>®</sup> Core<sup>™</sup> processor family PCH and discusses specific solutions for OS installation from a USB storage device with an OS that does not natively support xHCI in both the OEM and Channel Environments.

## 1.3 Overview

In prior generations, systems had both xHCI and EHCI controllers. In the case where a new OS install from a USB device was required the hardware would have the legacy EHCI drivers natively available in the OS. This native EHCI support allows the USB ports to function during a clean OS install thereby providing an end-user the ability to use a USB based storage or HID device to manually install the xHCI driver provided by Intel.

With the removal of the EHCI Controller, Intel does not anticipate any issues with only exposing an xHCI controller for Windows\* 8/8.1/10 or Linux\* (2.6.38 or later kernel) due to native xHCI support on the OS. However, Windows\* 7 does not natively support xHCI controller and relies on manual installation of the Intel's xHCI driver. While there are two types of USB pre-OS support for storage and HID (Human Interface Devices) devices, the concern with the lack of in-the-box support for the xHCI controller involves cases where the Windows\* 7 operating system needs to be installed from a USB storage device.

## 1.4 USB Storage Device Support

For the purpose of this document, USB Storage class devices are categorized into thumb drives and optical drives and represent the largest obstacle to the seamless removal of the legacy EHCI USB controllers. Windows\* 7 installations with xHCI hardware concerns become visible during OS installation and installation of the xHCI driver into a fresh OS.



## 1.5 xHCI-based Windows\* 7 Install Flow

The high-level flow of Windows\* 7 installation on a 'new' system from USB-attached storage consists of the following:

1. BIOS Discovery of Bootable Images - The bootable image is on a USB interface attached to the Intel xHCI controller.
2. BIOS Control Transferred to the Boot loader - The boot-loader uses an INT-13 DOS style handler (Legacy BIOS) or the Block IO Interface (UEFI BIOS) to access the USB device storage subsystem. During this phase, a virtual RAM drive is created and all fundamental OS information, including hardware drivers and WinPE kernel, are loaded into this virtual drive.
3. WinPE Kernel Starts - The USB specific issues start here as the xHCI driver is not available in the WinPE kernel. The BIOS is continuing to emulate storage services so that the boot loader is unaware that the physical storage device relies on the USB controller stack. When all of the drivers are loaded into the RAM drive, execution is handed off to the WinPE kernel. At this point in time, the WinPE kernel ceases to utilize the BIOS routines, and instead uses self-contained services within the RAM drive. The WinPE kernel will not be able to use the xHCI controller hardware as there is no xHCI driver in the RAM drive and access to the OS source files is severed. At this point, the install process is stuck, and a WinPE kernel message similar to the following will be displayed: "A required CD/DVD device driver is missing. If you have a driver floppy disk, CD, DVD, or USB flash drive, please insert now."
4. In addition to the primary issue of OS installation is the secondary issue of installing the xHCI driver, as the xHCI driver is not part of the Windows\* 7 installation package. Therefore, the xHCI drivers will not be copied onto the OS target drive, either HDD or SSD, etc. If the driver can be acquired via a network (wired or wireless), or some other non-USB interface, then the problem is resolved, and the xHCI controller will now be visible. However, if the platform is dependent on getting the xHCI driver from the OS install source path (USB-attached ODD or Key) then there is an issue.





## 2 xHCI USB Operating System Install Solutions

---

The following solutions are proposed for proper OS installation on xHCI only based systems:

1. Use Windows\* 8/8.1/10 and not Windows\* 7 – Windows\* 8/8.1/10 is a natively xHCI aware OS so the pre-install and post-install phases of the OS will have drivers for the xHCI hardware.
2. Use OEM provided DVD, special HD partition (E.g. Prebuilt Linux\* with native xHCI support); Network, or SATA-attached optical drives for OS installation. This option bypasses the issue, and allows the OS image to be installed from an interface that is natively supported by Windows\* 7. Once the baseline OS image is installed, the OS will be able to load the xHCI driver, where thereafter full USB functionality will be supported.
3. Instructions to create a Windows\* 7 Installation image with xHCI Driver included.
4. Instructions to create a custom Windows\* 7 WinPE image with the xHCI driver included – This is described in Boot From Customer WinPE with USB 3.0 Driver

### 2.1 Adding USB 3.0 Drivers to Windows\* 7 Install

1. Create a 'temp' folder under local HDD and copy all the files from Window 7 image to the folder  
E.g. c:\temp
2. Create a folder called 'wim' under c:\
3. Copy the install.wim and boot.wim located on the c:\temp folder to c:\wim  
The install.wim and boot.wim is located under c:\temp\source\
4. Download and install Windows Automated Installation Kit for Windows\* 7 OS.
5. Download the latest Intel® USB 3.0 eXtensible Host Controller Driver 4.x.x.x from Intel.com website.
6. Start AIK Deployment Tools Command Prompt using the "Deployment Tools Command Prompt" with administrator right
7. From the Command prompt change the target location to c:\wim  
`cd c:\wim`
8. Create a directory under c:\wim called 'mount'  
c:\wim\mount
9. Mount the WIM image with write permissions:  
E.g. `imagex /mountrw install.wim 5 c:\wim\mount`  
Imagex will tell which number is which Windows edition if it's not found.



**Note:** The Number 5 is in this case for Windows\* 7 Ultimate Edition 32 bit. Depending on the Windows\* 7 image, the number may be different. Look for the "Image Count" number display by imagex for the right number if there is error.

10. Add USB3.0 drivers to the install.wim image using DISM
  - a. Navigate to the location of the USB3.0 is located  
E.g. C:\USB3\x86
  - b. Add the USB3.0 Drivers (.inf) file to the install.wim image  
E.g. C:\USB3\x86>dism /image:C:\wim\mount /add-driver /driver:.  
/recurse
11. Save the WIM image with new drivers by unmounting and committing  
E.g. C:\USB3\x86>imagex /unmount /commit C:\wim\mount
12. Add USB3.0 driver to two installations of boot.wim with the following steps, so the drivers will be accessible during installation.
  - a. Add driver to boot.wim 1
    - i. Mount the boot.wim 1 image with write permissions  
E.g. C:\wim>imagex /mountrw boot.wim 1 c:\wim\mount
    - ii. Change to the directory where the USB3.0 drivers are located.  
E.g. C:\wim>cd C:\USB3\x86
    - iii. Add the USB3.0 Drivers (.inf) file to the base boot.wim 1 image  
E.g. C:\USB3\x86>dism /image:C:\wim\mount /add-driver  
/driver: /recurse
    - iv. Unmount the boot.wim 1 image  
E.g. C:\USB3\x86>imagex /unmount /commit C:\wim\mount
  - b. Add driver to boot.wim 2
    - i. Mount the boot.wim 2 image with write permissions  
E.g. C:\wim>imagex /mountrw boot.wim 2 c:\wim\mount
    - ii. Change to the directory where the USB3.0 drivers are located.  
E.g. C:\wim>cd C:\USB3\x86
    - iii. Add the USB3.0 Drivers (.inf) file to the base boot.wim 2 image  
E.g. C:\USB3\x86>dism /image:C:\wim\mount /add-driver  
/driver: /recurse
    - iv. Unmount the boot.wim 2 image  
E.g. C:\USB3\PPPT\x86>imagex /unmount /commit  
C:\wim\mount
13. Copy both install.wim and boot.wim from c:\wim to the c:\temp\source\ folder and replace the original files.



14. Burn files as new DVD (Make sure that the DVD is bootable) with favorite burning software (E.g. CDBurnerXP) or create image to a USB Flash Drive (UFD).

If using UFD, you can follow section [2.3](#) steps 2 to create bootable UFD and how to copy files to the UFD.

15. Install the Windows\* 7 as usual. After installation, Windows\* 7 will have the correct and working USB3.0 drivers.

**References:**

<http://superuser.com/questions/63773/how-to-add-drivers-to-windows-7-installation-dvd>

## 2.2 Boot from Customized WinPE with USB 3.0 Driver

1. Create a custom WinPE image
  - a. Start AIK Deployment Tools Command Prompt from Start menu
  - b. Copy the WinPE default to a working local directory path:  
cotype.cmd <arch> <destination>

Where <arch> can be x86, amd64, or ia64 and <destination> is a path to local directory. For example,

E.g. cotype.cmd x86 c:\winpe\_x86

The following directory structure will be created and all files copied

```
\Winpe_x86
\Winpe_x86\ISO
\Winpe_x86\Mount
```

- c. Mount the WinPE image with write permissions  
E.g. c:\winpe\_x86>imagex /mountrw winpe.wim 1 C:\winpe\_x86\mount
- d. Change to the directory where the USB3.0 drivers are located.  
E.g. c:\winpe\_x86>cd C:\USB3\x86
- e. Add the USB3.0 Drivers (.inf) file to the base WinPE image  
E.g. C:\USB3\x86>dism /image:C:\winpe\_x86\mount /add-driver /driver:./recurse
- f. Unmount the WinPE image  
C:\USB3\x86>imagex /unmount /commit C:\winpe\_x86\mount



- g. Copy the base image (Winpe.wim) into the \Winpe\_x86\ISO\sources folder, and rename the file to Boot.wim  
C:\USB3\x86>copy c:\winpe\_x86\winpe.wim  
c:\winpe\_x86\ISO\sources\boot.wim
2. Prepare the UFD
  - a. Open command prompt, use Diskpart to format the UFD as FAT32  
E.g.
    - i. Diskpart
    - ii. select disk 2
    - iii. clean
    - iv. create partition primary
    - v. select partition 1
    - vi. active
    - vii. format quick fs=fat32
    - viii. assign
    - ix. exit

This example above assumes Disk 2 is the UFD
3. Copy customized WinPE image to the UFD using command prompt  
xcopy c:\winpe\_x86\iso\\*.\*/e f:\  
Where f is the letter of the UFD
4. Copy USB driver to the same UFD where the customized WinPE is located
5. Boot from customized WinPE with USB3 driver added.
6. Navigate to the Windows\* 7 installation disk.  
E.g. E:
7. Start Windows\* 7 installation (USB ODD, USB Key or anywhere).  
E.g. Setup.exe
8. Choose custom installation
9. Choose load driver to select the following driver to be installed:  
"Intel(R) USB3.0 eXtensible Host Controller"  
"Intel(R) USB3.0 Root Hub"
10. Continue the installation as usual.
11. Newly installed Windows\* 7 will have the USB driver installed.

