

Intel® Server Utilities Deployment Procedure for Windows* Preinstallation Environment

White Paper

December 2009

Revision 5.0



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Intel® Server Utilities 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.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Intel, Pentium, Xeon and the Intel logo are trademarks or registered trademarks of Intel Corporation in the U. S. and other countries.

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

Copyright © 2009, Intel Corporation. All rights reserved.



Contents

1	Introduction	5
1.1	Purpose of This Document	5
1.2	Intended Audience	5
1.3	Prerequisites	5
1.4	Terminology	6
1.5	Conventions Used in this Document	7
2	Steps for Customized Windows*PE CD Preparation.....	8
2.1	For WinPE* 1.5 and 1.6.....	8
2.1.1	Step 1: Build Preparation	8
2.1.2	Step 2: Updating Windows* PE Registry for Intel Drivers.....	8
2.1.3	Step 3: Creating Customized Windows* PE	9
2.1.4	Step 4: Intel® Server Configuration Utilities and Driver Insertion.....	9
2.1.5	Step 5: Enabling WMI*	10
2.1.6	Step 6: Enabling Multi Processor Support.....	11
2.1.7	Step 7: CD Creation	11
2.2	For WinPE* 2.0 and 2.1.....	12
2.2.1	Prerequisites	12
2.2.2	Building WinPE* 2.x Customized Image	12
2.2.3	Creating a Bootable WinPE* 2.x CD, UFD and HDD.....	15
2.3	For WinPE* 3.0	17
2.3.1	Prerequisites	17
2.3.2	Building WinPE* 3.0 Customized Image.....	18
2.3.3	Creating a Bootable WinPE 3.0 CD, UFD and HDD	21
2.3.4	Capturing the Installation onto a Network Share	22
2.3.5	Deploying from a Network Share	22
3	Booting From Customized CD.....	24

Tables

Table 1: Prerequisites.....	6
Table 2: Terminology.....	6
Table 3: List of Packages to be installed with WinPE* 2.x	14
Table 4: Windows PE 3.0 optional packages	19



Revision History

Revision Number	Description	Revision Date
1.0	Initial release	April 2007
2.0	Added information on WinPE* 1.5, 2.0 ,and 2.1	May 2008
3.0	Updated WinPE* 2.0/2.1 with detailed steps to add Intel Utilities and Drivers.	January 2009
4.0	Updated Section 2.2, Building WinPE* 2.x Customized Image	March 2009
5.0	Added Section 2.3, Building WinPE* 3.0 Customized Image	December 2009



1 Introduction

1.1 Purpose of This Document

The purpose of this white paper is to provide basic guidance to prepare a customized Windows* Preinstallation Environment (Windows* PE) image, wherein *customization* refers to the inclusion of the Intel configuration and deployment Utilities and its drivers to the Windows* PE OS. A bootable Windows* PE CD is created from this customized image, from which the Intel Windows PE utilities -- iFlash32, SELViewer, SYSCFG, FRUSDR, and FWPIAUPD -- can be run without going through any other update procedure.

Steps to prepare the Intel® Server Configuration Utilities – Windows* PE CD are as follows:

1. Customize a Windows* PE OS to support Intel Utilities and drivers
2. Enable WMI and PNP support for Windows* PE OS
3. Create a bootable Windows* PE CD

1.2 Intended Audience

This document is intended for the following audience:

- OEM Customers who own a license to redistribute the customized Windows* PE CD with Server Configuration Utilities
- Windows System Administrators

1.3 Prerequisites

Customization of Windows* PE image to include Intel Server Configuration utilities and drivers requires the following:

- A computer running Microsoft Windows* XP or later versions.
- All files for the Intel Server Configuration utilities and drivers either available on the computer's local hard drive or network, or saved to removable media.



The following CDs are required for building WinPE* image

Table 1: Prerequisites

WinPE* Version	Build tools	Windows* OS
WinPE* 1.5	Windows* PE version 2004 CD containing build tools	Windows* XP with Service Pack 2 (SP2) CD
WinPE* 1.6	Windows* PE version 2005 CD containing build tools	Windows* Server 2003 with Service Pack 1 (SP1) CD
WinPE* 2.0	Windows Automated Installation Kit*(AIK) v1.0.0.0	Windows* Vista DVD
WinPE* 2.1	Windows Automated Installation Kit* (AIK) v1.0.0.0	Windows* Server 2008 DVD
WinPE* 3.0	Windows Automated Installation Kit* (AIK) for Windows 7	Windows* Server 2008 DVD

1.4 Terminology

Table 2: Terminology

Term	Definition
Windows* PE	The Microsoft Windows* Pre-installation Environment (Windows* PE) is a minimal Win32 subsystem with limited services, based on the Windows* kernel running in protected mode.
WMI*	Windows* Management Instrumentation (WMI*) is a scalable system management infrastructure that provides you with a standard way to interact with system management information.
PNP	Plug and Play (PNP) is a feature that enables user to plug in/out a device.
SCU	Intel® Server Configuration Utilities (SCU) are available in the 'Intel® Server Deployment ToolKit CD'.
OFU	One-Boot Flash Update (OFU) Utility is used to update the system BIOS, BMC (Base Board Management Controller) firmware, FRUSDR, HSC (Hot Swap Controller) firmware and LCP (Local Control Panel) firmware of server systems.
SYSCFG	System Configuration (SYSCFG) Utility is used to configure selective BIOS and firmware settings of server systems.
FRUSDR	FRUSDR Loader Utility is used to update systems' FRUs (Field Replaceable Unit) and SDRs (Sensor Data Record).
FWPIAUPD	Firmware and PIA Flash Update (FWPIAUPD) Utility is used for updating the boot block, operational code and Platform Information Area (PIA) for various micro-controllers on a given platform through the use of platform-specific Intel hex programming files.



IMB Driver	The Intelligent Management Bus (IMB) driver is the reference implementation of an IPMI specification for the Windows NT operating system. It is a driver that supports IPMB messaging to server management firmware on IPMI conformant servers.
IBSMUtil Driver	IBSMUtil Driver is used to access SMBIOS data.
Flash Update Driver	Flash Update Device Driver is used to update the BIOS (Basic Input/Output System) of a server.

1.5 Conventions Used in this Document

Typographic Convention

- Plain Text: The normal text typeface is used for ordinary text.
- **Bold**: In text, **Bold** typeface indicates the running head within a paragraph. In instances of command line execution, this indicates typical examples of the syntax of the various build tools.
- *Italic*: In text, an *Italic* typeface indicates the reference to 'directory' that is used in this document procedure.

Hardware Convention

- The CD-ROM drive letter is assigned to 'E:'.



2 Steps for Customized Windows* PE CD Preparation

2.1 For WinPE* 1.5 and 1.6

The Microsoft Windows* PE 1.5/1.6 image is modified to install Intel drivers for the execution of Intel utilities. This customization includes the support for the following:

- **PNP:** Adds Plug and Play device driver detection
- **WMI:** Adds the WMI providers and classes
- **Intel® Server Configuration Utilities:** Adds the binaries of Intel's Server Configuration Utilities
- **Intel Drivers:** Adds the support for the installation of three Intel drivers namely, IMB Driver, IBSMUtil Driver and Flash Update driver that are required by the Intel's Server Configuration utilities

The detailed procedure is explained in the following sections:

2.1.1 Step 1: Build Preparation

This step explains how to get the build tools that are necessary for the customization and creation of the Windows* PE image.

1. Place the '*Microsoft Windows* PE CD*' in the CD-ROM drive.
2. Create a directory on your machine to store the Windows PE build tools, denoted as *build_location*. For example, type: **md c:\build_x86**
3. Copy the contents of *cd_drive\Winpe* folder to *build_location*. For example, type: **xcopy e:\winpe c:\build_x86 /s**

2.1.2 Step 2: Updating Windows* PE Registry for Intel Drivers

This step describes the procedure to add the registry entries for the Intel drivers (IMB driver, IBSMUtil driver and Flash Update driver) to the Windows* PE registry.

1. Open the *c:\build_x86\WINPESYS.INF* file.
2. Search for "network configuration entries" section.



3. Copy the text in Appendix A and paste above the “network configuration entries” section.

Note: Please be careful to do the “Soft return” and “Hard return” when you copy and paste from PDF format Appendix to your c:\build_x86\WINPESYS.INF

(Soft return – Press the Enter key; Hard return – Press the keys Shift+Enter)

2.1.3 Step 3: Creating Customized Windows* PE

The ‘mkimg.cmd’ used in this procedure creates the Windows* PE image that has the required files for Windows* PE OS to run, along with the Intel drivers’ registry entries.

1. Remove the ‘Microsoft Windows PE CD’ from the CD-ROM drive and load the Windows OS CD.
2. Navigate to *build_location*. For example, type: **cd c:\build_x86**
3. Run the *Mking* command with the below additional switches.

```
mkimg.cmd <cd_drive> <winpe_image_dir> /PNP /WMI
```

For example, type:

```
mkimg.cmd e:\ c:\winpe /PNP /WMI
```

where, ‘e:\’ points to the Windows OS CD and

‘c:\winpe’ is the destination directory where the customized Windows PE image files are copied.

‘/PNP’ and ‘/WMI’ switches provides the support for ‘Plug and Play’ and ‘Windows Management Instrumentation (WMI)’ respectively.

Note: As “security.log” cannot be successfully copied, the last messages could be “Failed to copy all the required files for WinPE”. This will not affect customized WinPE* CD building.

2.1.4 Step 4: Intel® Server Configuration Utilities and Driver Insertion

This step describes the addition of Intel’s Server Configuration utilities and driver binaries to the customized Windows* PE image.

1. Ensure that all files for the Intel® Server Configuration utilities are accessible. If necessary, load the removable media that contains the files.
2. Run the ‘c:\build_x86\drvinst.exe’ command to add Intel driver (IMB, IBSMUtil and Flash update) files to the Windows PE image:

```
drvinst.exe /inf: source_directory <winpe_image_dir>
```

where:
<source_folder> is the folder that holds the Intel driver files:
<winpe_image_dir> is the destination folder into which the customized



Windows PE* image files are copied.

For example: **c:\build_x86\drvinst.exe /inf:c:\Utility_Drivers c:\winpe**
where 'c:\Utility Drivers' contain all of the files for the Intel Server Configuration Utilities including:

- IMB driver: imbdrv.inf, imbdrv.sys
- IBSMUtil driver: ibsmutil.inf, ibsmutil.sys
- Flash Update driver: flashud.inf, FLASHUD.sys

This adds Intel driver files (IMB, IBSMUtil and Flash update) to the Windows PE image.

Note: FRUSDR, FWPIAUPD, IFLASH, SELVIEWER and SYSCFG utility for WinPE* may come with different version IMB driver, IBSMUtil driver and Flash Update driver. Please check with time stamp and copy the latest .inf and .sys files into c:\Utility_Drivers from FRUSDR, FWPIAUPD, IFLASH, SELVIEWER and SYSCFG utility for WinPE*. Also, you have to be careful as the drivers are not the same between x64 and x86.

3. Create a folder under *<winpe_image_dir>* and copy the Intel® Server Configuration Utility files into this folder.

md c:\winpe\IntelUtils

xcopy c:\Utilfiles c:\winpe\IntelUtils /s

where, 'c:\Utilfiles' corresponds to the folder that contains all of the files for the BIOS & Firmware Update & Configuration Utilities.

Note 1: Download respective platform's BIOS & Firmware Update & Configuration Utilities for Windows PE* from support web site and unzip it to one folder. Copy all utilities (including directory) that you planned to use under "c:\winpe\IntelUtils".

Note 2: If you want to include use customized WinPE CD for update system's BIOS/BMC/FRUSDR/HSC. You can also copy BIOS/BMC/FRUSDR/HSC image or hex files into "c:\winpe\IntelUtils" as you like.

2.1.5 Step 5: Enabling WMI *

The below mentioned extra files/folders need be copied from 'Microsoft Windows PE CD' for the starting WMI service provider.

1. Load the *Microsoft Windows* PE CD* in the CD-ROM drive.
2. Create below mentioned empty directories under *<winpe_image_dir>\i386\system32\wbem* folder
For example:
md C:\winpe\I386\SYSTEM32\WBEM\REPOSITORY
md C:\winpe\I386\SYSTEM32\WBEM\AUTORECOVER
md C:\winpe\I386\SYSTEM32\WBEM\LOGS
md C:\winpe\I386\SYSTEM32\WBEM\MOF
3. Copy the contents of 'REPOSITORY' folder under '*<cdrom_drive>:\I386\SYSTEM32\WBEM\REPOSITORY*' to the customized *<winpe_image_dir>* folder.
For example:
xcopy e:\I386\SYSTEM32\WBEM\REPOSITORY*.*
C:\winpe\I386\system32\wbem\REPOSITORY /s



2.1.6 Step 6: Enabling Multi Processor Support

This section enables the Multi Processor support for Windows* PE OS.

1. Delete the Uni processor kernel exe ('ntoskrnl.exe') in the `<winpe_image_dir>\i386\system32` folder.
For example:
del c:\winpe\i386\SYSTEM32\ntoskrnl.exe
2. Rename the Multi processor kernel 'ntkrnlmp.exe' to 'ntoskrnl.exe'.
For example:
`ren c:\winpe\i386\SYSTEM32\ntkrnlmp.exe ntoskrnl.exe`
3. Open the 'txtsetup.sif' file that is present in `<winpe_image_dir>\i386\` folder.
For example:
`notepad C:\winpe\i386\txtsetup.sif`
4. Search for '[Hal.Load]' section.
Replace:
'mps_mp = halapic.dll' with 'mps_mp = halmmps.dll'
'acpiapic_mp = halaacpi.dll' with 'acpiapic_mp = halmacpi.dll'.

2.1.7 Step 7: CD Creation

`<winpe_image_dir>` folder that was created in the above steps contains the customized Windows PE OS files along with the Intel utility binaries and drivers. This step explains the usage of 'oscdimg.exe' for the creation of customized Windows* PE '.iso' image

1. Run the '**c:\build_x86\oscdimg.exe**' command with these options to create the .iso image:
`oscdimg -b<boot_file_location> -n <winpe_image_dir> <iso_image_name>`
For example,
c:\build_x86\oscdimg -bc:\build_x86\etfsboot.com -n c:\winpe c:\winpex86.iso
where:
'c:\winpe' folder contains the customized Windows PE files
'c:\winpex86.iso' is the name of the ISO image that 'oscdimg' creates.
2. Use the CD-recording software to burn the ISO image file to a blank CD.

This customized CD can be directly used to boot the target system with Windows PE OS.



2.2 For WinPE* 2.0 and 2.1

This section helps create a customized WinPE 2.x image from Windows* AIK image as detailed in the following sections:

Prerequisites

Building customized WinPE* 2.x Image.

Creating a Bootable WinPE* 2.x CD-ROM, USB flash drive (UFD) and HDD.

2.2.1 Prerequisites

Following is the prerequisites for building a Windows PE* build image:

- Install the Microsoft Windows* Automated Installation Toolkit (AIK)

Following are the prerequisites for starting Windows PE* using a RAM disk:

- An x86 or amd64-based computer.
- A bootable Windows PE* image created from a Windows PE* 2.x base image.
- A minimum of 1 gigabyte (GB) of RAM when using the default Windows PE disk image.

Note: If you use a customized RAM disk image, the computer's RAM must be at least 100 MB more than the size of the customized RAM disk image.

2.2.2 Building WinPE* 2.x Customized Image

This section contains steps to build a custom WinPE 2.x image.

The first step in creating a customized Windows PE 2.x image is to modify the base Windows PE image (Winpe.wim) by using ImageX and PEImg tools. ImageX is required to mount the files to a local directory. PEImg enables you to add or remove packages, and to add out-of-box drivers and language packs. ImageX then enables you to recapture changes back into a .wim file.

Step 1: Build a Windows PE Build environment using AIK.

After installing WAIK, open WinPE* Tools command prompt, go to **Start>>All Programs>>Microsoft Windows AIK>>Windows PE Tools Command Prompt.**

At the command prompt, run the Copype.cmd script. The script requires two arguments: hardware architecture and destination location, where architecture can be **x86** or **amd64** and destination is a path to local directory. For example,

Format: copype <architecture> <destination folder>

```
copype.cmd x86 c:\winpe2\x86
```



Note: All the examples continued are for creating x86 based WinPE 2.x image. You can change x86 to amd64 for building an Intel64 WinPE 2.x image.

The script creates the following directory structure and copies all the necessary files for that architecture. For Example,

```
\winpe2\x86  
\winpe2\x86\ISO  
\winpe2\x86\mount
```

Your current working directory will now be **"c:\winpe2\x86"**

Step 2: Create a custom WinPE 2.x Image.

Mount the base Windows PE* image (Winpe.wim) to the \mount directory by using ImageX. For example,

Format : `imagex /mountrw image_file image_number image_name image_path`

`imagex /mountrw c:\winpe2\x86\winpe.wim 1 c:\winpe2\x86\mount`

where,

image_file - The path of the WIM file containing the specified image.

image_number - The number that identifies the image within the WIM file.

image_name - The name that identifies the image within the WIM file.

image_path - The path where the specified image will be mounted.

Add all required Windows* features to the base image by using the peimg /install command. For example,

Format: `PEIMG {Command} {Image} [Flags]`

`peimg /install=WinPE-HTA-Package c:\winpe2\x86\mount\Windows`

where first parameter is the peimg command 'install' with the package name to be installed and second parameter is the path to the Windows directory within the base Windows PE image.



Repeat the same steps for all the packages mentioned in following table:

Table 3: List of Packages to be installed with WinPE* 2.x

Package Name	Description
WinPE-HTA-Package	HTML Application support
WinPE-MDAC-Package	Microsoft* Data Access Component support
WinPE-Scripting-Package	Windows* Script Host support
WinPE-SRT-Package	Windows* Recovery Environment support
WinPE-XML-Package	Microsoft* XML (MSXML) Parser support
WinPE-WMI-Package	Windows* Management Instrumentation support

Verify that the packages were installed by using the `peimg /list` command to view all packages in the current image. For example,

```
peimg /list c:\winpe2\x86\mount\windows
```

where,

first parameter is the `peimg` command 'list' and second parameter is the path to the Windows directory within the base Windows PE* image.

Note: - All the inserted packages will show ('+') in INS column.

Create a folder named WinPEUtils and add the required utilities and programs to the Windows PE* image, For example

```
md c:\winpe2\x86\mount\WinPEUtils
```

```
xcopy c:\WinPEUtils\* c:\winpe2\x86\mount\WinPEUtils /S
```

Step 3: Prepare a Windows PE* Image

- Now prepare the WinPE* image for deployment.
peimg /prep c:\winpe2\x86\mount\windows
where,
/prep optimizes the WinPE image for size.



- Commit changes to the image and unmount
Format: `IMAGEX /UNMOUNT [/COMMIT] [image_path]`
imagex /unmount c:\winpe2\x86\mount /commit
where,
'c:\winpe2\x86\mount' is the path to be unmounted
/commit is the flag to save changes to the mounted WIM file.
- Replace the default WinPE2.1 image
copy c:\winpe2\x86\winpe.wim
c:\winpe2\x86\ISO\sources\boot.wim /Y

2.2.3 Creating a Bootable WinPE* 2.x CD, UFD and HDD

Step 1: Creating a Bootable WinPE* 2.x CD-ROM

Format: `oscdimg -n -b <boot_file_location> <winpe_image_dir> <iso_image_name>`

- **oscdimg -n -bc:\winpe2\x86\etfsboot.com c:\winpe2\x86\ISO**
c:\winpe2x86.iso
where
'c:\winpe2\x86\ISO' folder contains the customized Windows PE files and
'c:\winpe2x86.iso' is the name of the ISO image that 'oscdimg' creates.

Step 2: Create a bootable UFD device

- Connect the UFD to a machine running Windows* 2008 or Windows PE* image
- Type "diskpart" from command prompt.
- Follow below sequence of commands to create a bootable UFD, assumption is that user understands the correct partition assigned to the UFD device based on size of UFD device. In this example it is Disk 1.
diskpart>list disk
diskpart>select disk 1
diskpart>clean
diskpart>create partition primary
diskpart>select partition 1
diskpart>active
diskpart>format fs=fat32
diskpart>assign
diskpart>exit

Caution:- selecting a wrong disk can wipe out entire data.

- Connect UFD back to the WinPE build machine.
- You can copy the WinPE* image to the bootable UFD device.
xcopy c:\winpe2\x86\iso*. * /s /e /f f:
where
f: is the drive assigned in the computer where you build the WinPE2.1 image.



Step 3: Create a bootable HDD device.

- On the deployment sever, insert your Windows PE media and restart the computer. You can use the step1 (Creating a useable WinPE* 2.1 CR ROM) to create a WinPE* media.
- Type "diskpart" from a windows 2008 or Windows PE image.

Follow the below sequence of commands, assumption the user understands which is the correct partition assigned to the hard disk based on size of device. In this example it is Disk 0.

```
diskpart>list disk  
diskpart>select disk 0  
diskpart>clean  
diskpart>create partition primary size=350  
diskpart>select partition 1  
diskpart>active  
diskpart>format  
diskpart>exit
```

Caution:- selecting a wrong disk can wipe out entire data.

- Once you have a formatted, active partition, you can copy the Windows PE source files from your bootable media to the hard disk. A Windows PE RAM disk requires the following directory structure.

```
\boot  
\sources
```

- You can manually create the directory structure or use the xcopy command to automatically build and to copy the appropriate files from your bootable media. At a command prompt, type:

```
xcopy d:\*.* /s /e /f c:\
```

where ,

d is the letter of your CD-ROM drive or UFD media and c is the letter of your hard disk.



2.3 For WinPE* 3.0

This section helps create a customized WinPE 3.0 image from Windows* AIK image as detailed in the following sections:

Prerequisites

Building WinPE* 3.0 Customized Image

Creating a Bootable WinPE* 2.x CD, UFD and HDD

Capturing the Installation onto a Network Share

Deploying from a Network Share

2.3.1 Prerequisites

Following is the prerequisites for building a Windows PE* build image:

- Install the Microsoft Windows* Automated Installation Toolkit (AIK)
- Install .Net framework 2.0, which is a system requirement for the Windows* AIK
- Install Microsoft* Core XML Services (MSXML) 6 SP1. This service should be installed only when running Windows server 2003 SP1.

Following are the prerequisites for starting Windows PE* using a RAM disk:

- An x86, amd64 or ia64-based computer.
- A bootable Windows PE* image created from a Windows PE* 3.0 base image.
- A minimum of 1 gigabyte (GB) of RAM when using the default Windows PE disk image.

Note: *If you use a customized RAM disk image, the computer's RAM must be at least 100 MB more than the size of the customized RAM disk image.*



2.3.2 Building WinPE* 3.0 Customized Image

This section provides a step-to-step guide to build a WinPE* 3.0 image.

The first step in creating a customized Windows® PE 3.0 image is to modify the base Windows PE image (winpe.wim) by using the Deployment Image Servicing and Management (DISM) tool. DISM extracts the files to a local directory and enables you to add and remove packages (optional components and language packs). In addition, you can add out-of-box drivers. DISM provides the same mounting and unmounting operations as ImageX.

Step 1: Prepare and Cleanup

After installing WAIK, go to **Start >> All Programs >> Windows AIK >> Deployment Tools Command Prompt**. The menu shortcut opens a Command Prompt window and automatically sets environment variables to point to all the necessary tools. By default, all tools are installed at C:\Program Files\Windows AIK\Tools.

At the Deployment Tools Command Prompt, run the script.

Format: Dism /cleanup-wim

This command will delete all of the resources associated with a mounted WIM image that has been abandoned. It will not unmount currently mounted images, nor will it delete images that can be remounted.

Step 2: Build a Windows PE Build environment

Format: copy.cmd <architecture> <destination>

Where <architecture> can be x86, amd64, or ia64, it depends on the architecture of your computer. And <destination> is a path to a local directory.

For example: copy.cmd x86 c:\winpe_x86

The script creates the following directory structure and copies all the necessary files for that architecture. For example,

```
\winpe_x86
\winpe_x86\ISO
\winpe_x86\mount
```

Note: All the examples continued are for creating x86 based WinPE 3.0 image. You can change x86 to amd64 for building an Intel64 WinPE 3.0 image or ia64 for building an Itanium64 WinPE 3.0 image.

Step 3: Mounting WIM file on your local drive

Mount the base Windows PE* image (Winpe.wim) to the \mount directory:



Format:

```
Dism /Mount-wim /WimFile:<Path_to_image.wim> /index:<image_index>  
/MountDir:<path_to_mount_directory>
```

For example:

```
Dism /Mount-Wim /WimFile:c:\winpe_x86\winpe.wim /index:1  
/MountDir:c:\winpe_x86\mount
```

Step 4: Adding packages to your WIM

Format:

```
Dism /image:<path_to_mount_directory> /Add-Package  
PackagePath:"<path_to_package>"
```

By default, all packages are at c:\Program Files\Windows
AIK\Tools\PETools\<architecture>\WinPE_FPs\

For example:

```
Dism /image:C:\winpe_x86\mount /Add-Package  
/PackagePath:"C:\Program Files\Windows  
AIK\Tools\PETools\x86\WinPE_FPs\winpe-wmi.cab"
```

Windows PE 3.0 provides the following optional packages:

Table 4: Windows* PE 3.0 optional packages

Package Name	Description
WinPE-FONTSupport- <region>	Additional font support.
WinPE-HTA	HTML Application support.
Winpe-LegacySetup	The Media Setup feature package.
WinPE-MDAC	Microsoft® Data Access Component support.
Package Name	Description
WinPE-PPPoE	Enables Point-to-Point Protocol over Ethernet (PPPoE) support.
WinPE-Scripting	Windows Script Host (WSH) support.
WinPE-Setup	The Setup feature package (parent).
WinPE-Setup-Client	The Client Setup feature package (child).
WinPE-Setup-Server	The Server Setup feature package (child).
WinPE-WMI	Windows Management Instrumentation (WMI) support.
WinPE-WDS-Tools	The Windows Deployment Services tools feature package.

Note:



- You must use the DISM tool to install the appropriate packages. The packages are located in the directory
`\Windows AIK\Tools\PETools\`
- The winpe-xml package is now included in the base boot.wim.
- [Since the base image has MSIPMI driver installed by default, insert the "winpe-wmi.cab" so that WMI services are available in the WinPE image](#)

Step 5: Unmount your finished image and commit the changes

Format:

```
Dism /Unmount-wim /MountDir:<path_to_mount_directory> /Commit
```

For example:

```
Dism /Unmount-Wim /MountDir:C:\winpe_x86\mount\ /Commit
```

Step 6: Copy your custom image into \ISO\sources folder and rename to boot.wim.

Format:

```
copy <path_to_image.wim> <path_to_boot.wim> /Y
```

For example:

```
copy c:\winpe_x86\winpe.wim c:\winpe_x86\ISO\sources\boot.wim /Y
```

Step 7: Create your customize image of WinPE* 3.0

Create a Windows PE image (.iso) file by using the Oscdimg tool from the Deployment Tools Command Prompt.

Format:

```
oscdimg -n -b<boot_file_location> <winpe_image_dir> <iso_image_name>
```

For example:

```
oscdimg -n -bc:\winpe_x86\etfsboot.com c:\winpe_x86\ISO  
c:\winpe_x86\winpe_x86.iso
```



2.3.3 Creating a Bootable WinPE 3.0 CD, UFD and HDD

Creating a bootable CD

Burn the image (Winpe_x86.iso) to a CD-ROM disc. Windows AIK does not include CD-ROM burning software. Use trusted third-party software to burn the image to a CD-ROM disc.

Creating a bootable UFD

- Connect the UFD to a machine running Windows* 2008 or Windows PE* image
- Type "diskpart" from command prompt.
- Follow below sequence of commands to create a bootable UFD, assumption is that user understands the correct partition assigned to the UFD device based on size of UFD device. In this example it is Disk 1.

```
diskpart
diskpart>list disk
diskpart>select disk 1
diskpart>clean
diskpart>create partition primary
diskpart>select partition 1
diskpart>active
diskpart>format fs=fat32
diskpart>assign
diskpart>exit
```

Caution:- selecting a wrong disk can wipe out entire data.

- Connect UFD back to the WinPE build machine.
- You can copy the WinPE* image to the bootable UFD device.
xcopy c:\winpe_x86\iso*. * /s /e /f f:
Where f: is the drive assigned in the computer where you build the WinPE 3.0 image.

Creating a bootable HDD device

- On the deployment sever, insert your Windows PE media and restart the computer.
- Type "diskpart" from a Windows 2008 or Windows PE image.
- Follow the below sequence of commands, assumption the user understands which is the correct partition assigned to the hard disk based on size of device. In this example it is Disk 0.

```
diskpart
```



```
diskpart>list disk
diskpart>select disk 0
diskpart>clean
diskpart>create partition primary size=350
diskpart>select partition 1
diskpart>active
diskpart>format
diskpart>exit
```

Caution:- selecting a wrong disk can wipe out entire data.

- Once you have a formatted, active partition, you can copy the Windows PE source files from your bootable media to the hard disk. A Windows PE RAM disk requires the following directory structure.
boot
sources
- You can manually create the directory structure or use the xcopy command to automatically build and to copy the appropriate files from your bootable media. At a command prompt, type:
xcopy d:*.* /s /e /f c:
Where d: is the letter of your CD-ROM drive or UFD media and c: is the letter of your hard disk.

2.3.4 Capturing the Installation onto a Network Share

1. On your reference computer, insert your Windows PE media and restart the computer.
2. Capture an image of the reference installation by using the ImageX tool located on your Windows PE media.
For example:
e:\imagex.exe /capture d: d:\myimage.wim "my Install" /compress fast /verify
Where e: is the letter of your CD-ROM
3. Copy the image to a network location. Windows PE provides network support.
net use y: \\network_share\images
copy d:\myimage.wim y:
If necessary, provide network credentials for appropriate network access.

2.3.5 Deploying from a Network Share

After you have an image of your reference installation, you can deploy the image onto new hardware.

In this step, you use the DiskPart tool to format the hard drive, and then you copy an image from the network share.

1. On your destination computer, insert your Windows PE media and boot into Windows PE media.



2. Format the hard drive to reflect the disk configuration requirements by using the DiskPart tool from the Windows PE Command Prompt window.

For example:

```
diskpart  
diskpart>select disk 0  
diskpart>clean  
diskpart>create partition primary size=200  
diskpart>select partition 1  
diskpart>format fs=ntfs label="system"  
diskpart>assign letter=c  
diskpart>active  
diskpart>create partition primary  
diskpart>select partition 2  
diskpart>format fs=ntfs label="Windows"  
diskpart>assign letter=d  
diskpart>exit
```

Note: You can create a script with this information by saving it to a text file and by storing it in the same location as your image. To run the script from a Windows PE Command Prompt Window, type:

```
diskpart /s <scriptname>.txt
```

where <scriptname> is the name of the text file that includes the Diskpart commands.

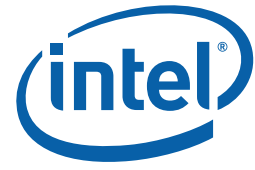
3. Copy the image from the network share to your local hard drive. For example, at a command prompt, type
net use y: \\network_share\images
copy y:\myimage.wim d:
If necessary, provide network credentials for appropriate access.
4. Apply the image to the hard drive by using the ImageX tool located on your Windows PE media. For example, at a command prompt, type
e:\imagex.exe /apply d:\myimage.wim 1 d:
5. Use BCDboot to initialize the Boot Configuration Data (BCD) store and copy boot environment files to the system partition. For example, at a command prompt, type
d:\windows\system32\bcdboot d:\windows



3 *Booting From Customized CD*

Insert the above-created CD into the computer on which you want to run Windows PE and start the computer. The computer will start Windows PE OS from the CD, which is a **read** only media. Once the OS is up, the user can navigate to the Intel's system configuration utilities repository and run the utilities.

Note: For WinPE 1.x, plug and play hardware detection occurs only while the Factory.exe - Windows PE command runs. Therefore, the Plug and Play devices (like the USB device) that you want to add to your Windows* PE image must be attached to the computer when you start Windows* PE.



Appendix A - Adding Intel Drivers in winpesys.inf

```
;
; Intel drivers
;

HKLM,"SYSTEM\ControlSet001\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0001","InfPath",REG_SZ,"oem3.inf"

HKLM,"SYSTEM\ControlSet001\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0001","InfSection",REG_SZ,"IMBDrvInstallSection"

HKLM,"SYSTEM\ControlSet001\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0001","ProviderName",REG_SZ,"Intel, Inc."

HKLM,"SYSTEM\ControlSet001\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0001","DriverDateData",REG_BINARY,00,40,cb,58,e8,de,c3,01

HKLM,"SYSTEM\ControlSet001\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0001","DriverDate",REG_SZ,"1-20-2004"

HKLM,"SYSTEM\ControlSet001\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0001","DriverVersion",REG_SZ,"6.2.0.0"

HKLM,"SYSTEM\ControlSet001\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0001","MatchingDeviceId",REG_SZ,"*imbdrv"

HKLM,"SYSTEM\ControlSet001\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0001","DriverDesc",REG_SZ,"Intel Intelligent Management Bus Driver V8.10"

HKLM,"SYSTEM\ControlSet001\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0002","InfPath",REG_SZ,"oem4.inf"
```



HKLM,"SYSTEM\\ControlSet001\\Control\\Class\\{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0002","InfSection",REG_SZ,"Flashud_Inst"

HKLM,"SYSTEM\\ControlSet001\\Control\\Class\\{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0002","InfSectionExt",REG_SZ, ".NT"

HKLM,"SYSTEM\\ControlSet001\\Control\\Class\\{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0002","ProviderName",REG_SZ,"Intel, Inc."

HKLM,"SYSTEM\\ControlSet001\\Control\\Class\\{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0002","DriverDateData",REG_BINARY,00,c0,bf,b6,30,c4,c3,01

HKLM,"SYSTEM\\ControlSet001\\Control\\Class\\{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0002","DriverDate",REG_SZ,"12-17-2003"

HKLM,"SYSTEM\\ControlSet001\\Control\\Class\\{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0002","DriverVersion",REG_SZ,"1.0.0.0"

HKLM,"SYSTEM\\ControlSet001\\Control\\Class\\{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0002","MatchingDeviceId",REG_SZ,"*int0800"

HKLM,"SYSTEM\\ControlSet001\\Control\\Class\\{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0002","DriverDesc",REG_SZ,"Intel 28F320C3 Flash Update Device Driver"

HKLM,"SYSTEM\\ControlSet001\\Control\\Class\\{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0003","InfPath",REG_SZ,"ibsmutil.inf"

HKLM,"SYSTEM\\ControlSet001\\Control\\Class\\{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0003","InfSection",REG_SZ,"Util_Device_Inst"

HKLM,"SYSTEM\\ControlSet001\\Control\\Class\\{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0003","InfSectionExt",REG_SZ, ".NTx86"

HKLM,"SYSTEM\\ControlSet001\\Control\\Class\\{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0003","ProviderName",REG_SZ,"Intel Corporation"

HKLM,"SYSTEM\\ControlSet001\\Control\\Class\\{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0003","DriverDateData",REG_BINARY,00,80,f0,fc,32,88,c6,01



HKLM,"SYSTEM\ControlSet001\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0003","DriverDate",REG_SZ,"6-5-2006"

HKLM,"SYSTEM\ControlSet001\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0003","DriverVersion",REG_SZ,"1.0.0.0"

HKLM,"SYSTEM\ControlSet001\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0003","MatchingDeviceId",REG_SZ,"*intelutildev"

HKLM,"SYSTEM\ControlSet001\Control\Class\{4D36E97D-E325-11CE-BFC1-08002BE10318}\0003","DriverDesc",REG_SZ,"Intel Server Management Utility Device v1.0"

HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0001","ClassGUID",REG_SZ,"{4D36E97D-E325-11CE-BFC1-08002BE10318}"

HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0001","Class",REG_SZ,"System"

HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0001","HardwareID",REG_MULTI_SZ,"2a,00,49,00,4d,00,42,00,44,00,52,00,56,00,00,00,00,00"

HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0001","Driver",REG_SZ,"{4D36E97D-E325-11CE-BFC1-08002BE10318}\0001"

HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0001","Mfg",REG_SZ,"Intel, Inc."

HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0001","Service",REG_SZ,"imbdrv"

HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0001","DeviceDesc",REG_SZ,"Intel Intelligent Management Bus Driver V8.10"

HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0001","ConfigFlags",REG_DWORD,00000000

HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0001","Capabilities",REG_DWORD,00000000

HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0001\LogConf"

HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0001\Control","ActiveService",REG_SZ,"imbdrv"



```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0002","ClassGUID",REG_SZ,"{4D36E97D-E325-11CE-BFC1-08002BE10318}"
```

```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0002","Class",REG_SZ,"System"
```

```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0002","HardwareID",REG_MULTI_SZ,"2a,00,49,00,4e,00,54,00,30,00,38,00,30,00,30,00,00,00,00,00"
```

```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0002","Driver",REG_SZ,"{4D36E97D-E325-11CE-BFC1-08002BE10318}\0002"
```

```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0002","Mfg",REG_SZ,"Intel, Inc."
```

```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0002","Service",REG_SZ,"int0800"
```

```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0002","DeviceDesc",REG_SZ,"Intel I 28F320C3 Flash Update Device Driver"
```

```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0002","ConfigFlags",REG_DWORD,00000000
```

```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0002","Capabilities",REG_DWORD,00000000
```

```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0002\LogConf"
```

```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0002\Control","ActiveService",REG_SZ,"int0800"
```

```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0003","ClassGUID",REG_SZ,"{4D36E97D-E325-11CE-BFC1-08002BE10318}"
```

```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0003","Class",REG_SZ,"System"
```

```
HKLM,"SYSTEM\ControlSet001\Enum\Root\SYSTEM\0003","HardwareID",REG_MULTI_SZ,"2a,00,49,00,6e,00,74,00,65,00,6c,00,55,00,74,00,69,00,6c,00,44,00,65,00,56,00,00,00,00,00"
```



HKLM,"SYSTEM\\ControlSet001\\Enum\\Root\\SYSTEM\\0003","Driver",REG_SZ,"{4D36E97D-E325-11CE-BFC1-08002BE10318}\\0003"

HKLM,"SYSTEM\\ControlSet001\\Enum\\Root\\SYSTEM\\0003","Mfg",REG_SZ,"Intel Corporation"

HKLM,"SYSTEM\\ControlSet001\\Enum\\Root\\SYSTEM\\0003","Service",REG_SZ,"IBSMUTIL"

HKLM,"SYSTEM\\ControlSet001\\Enum\\Root\\SYSTEM\\0003","DeviceDesc",REG_SZ,"Intel Server Management Utility Device v1.0"

HKLM,"SYSTEM\\ControlSet001\\Enum\\Root\\SYSTEM\\0003","ConfigFlags",REG_DWORD,00000000

HKLM,"SYSTEM\\ControlSet001\\Enum\\Root\\SYSTEM\\0003","Capabilities",REG_DWORD,00000000

HKLM,"SYSTEM\\ControlSet001\\Enum\\Root\\SYSTEM\\0003\\LogConf"

HKLM,"SYSTEM\\ControlSet001\\Enum\\Root\\SYSTEM\\0003\\Device Parameters"

HKLM,"SYSTEM\\ControlSet001\\Enum\\Root\\SYSTEM\\0003\\Control","ActiveService",REG_SZ,"IBSMUTIL"

HKLM,"SYSTEM\\ControlSet001\\Services\\imbdrv","Type",REG_DWORD,00000001

HKLM,"SYSTEM\\ControlSet001\\Services\\imbdrv","Start",REG_DWORD,00000002

HKLM,"SYSTEM\\ControlSet001\\Services\\imbdrv","ErrorControl",REG_DWORD,00000001

HKLM,"SYSTEM\\ControlSet001\\Services\\imbdrv","ImagePath",REG_EXPAND_SZ,"System32\\Drivers\\imbdrv.sys"

HKLM,"SYSTEM\\ControlSet001\\Services\\imbdrv","DisplayName",REG_SZ,"Intel Intelligent Management Bus Driver V8.10"

HKLM,"SYSTEM\\ControlSet001\\Services\\imbdrv","Configuration",REG_DWORD,00000011

HKLM,"SYSTEM\\ControlSet001\\Services\\imbdrv","DiagCode",REG_DWORD,0000007f



HKLM,"SYSTEM\\ControlSet001\\Services\\imbdrv","InstallRefCount",REG_DWORD,00000001

HKLM,"SYSTEM\\ControlSet001\\Services\\imbdrv\\Security","Security",REG_BINARY,01,00,14,80,b8,00,00,00,c4,00,00,00,14,00,00,00,30,00,00,00,02,00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,00,00,02,00,88,00,06,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,05,20,00,00,00,20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,04,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,06,00,00,00,00,14,00,00,01,00,00,01,01,00,00,00,00,05,0b,00,00,00,00,00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,01,01,00,00,00,00,05,12,00,00,00

HKLM,"SYSTEM\\ControlSet001\\Services\\imbdrv\\Enum","0",REG_SZ,"ROOT\\SYSTEM\\0001"

HKLM,"SYSTEM\\ControlSet001\\Services\\imbdrv\\Enum","Count",REG_DWORD,00000001

HKLM,"SYSTEM\\ControlSet001\\Services\\imbdrv\\Enum","NextInstance",REG_DWORD,00000001

HKLM,"SYSTEM\\ControlSet001\\Services\\int0800","Type",REG_DWORD,00000001

HKLM,"SYSTEM\\ControlSet001\\Services\\int0800","Start",REG_DWORD,00000003

HKLM,"SYSTEM\\ControlSet001\\Services\\int0800","ErrorControl",REG_DWORD,00000001

HKLM,"SYSTEM\\ControlSet001\\Services\\int0800","ImagePath",REG_EXPAND_SZ,"System32\\Drivers\\flashud.sys"

HKLM,"SYSTEM\\ControlSet001\\Services\\int0800","DisplayName",REG_SZ,"Intel 28F320C3 Flash Update Device Driver"

HKLM,"SYSTEM\\ControlSet001\\Services\\int0800","Group",REG_SZ,"System Bus Extender"

HKLM,"SYSTEM\\ControlSet001\\Services\\int0800","InstallRefCount",REG_DWORD,00000001



HKLM,"SYSTEM\ControlSet001\Services\int0800\Security","Security",REG_BINARY,01,00,14,80,b8,00,00,00,c4,00,00,00,14,00,00,00,30,00,00,00,02,00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,00,00,02,00,88,00,06,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,04,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,06,00,00,00,00,14,00,00,01,00,00,01,01,00,00,00,00,00,05,0b,00,00,00,00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

HKLM,"SYSTEM\ControlSet001\Services\int0800\Enum","0",REG_SZ,"ROOT\SYSTEM\0002"

HKLM,"SYSTEM\ControlSet001\Services\int0800\Enum","Count",REG_DWORD,00000001

HKLM,"SYSTEM\ControlSet001\Services\int0800\Enum","NextInstance",REG_DWORD,00000001

HKLM,"SYSTEM\ControlSet001\Services\IBSMUTIL","Type",REG_DWORD,00000001

HKLM,"SYSTEM\ControlSet001\Services\IBSMUTIL","Start",REG_DWORD,00000003

HKLM,"SYSTEM\ControlSet001\Services\IBSMUTIL","ErrorControl",REG_DWORD,00000001

HKLM,"SYSTEM\ControlSet001\Services\IBSMUTIL","Tag",REG_DWORD,00000006

HKLM,"SYSTEM\ControlSet001\Services\IBSMUTIL","ImagePath",REG_EXPAND_SZ,"System32\Drivers\ibsmutil.sys"

HKLM,"SYSTEM\ControlSet001\Services\IBSMUTIL","Group",REG_SZ,"Extended Base"

HKLM,"SYSTEM\ControlSet001\Services\IBSMUTIL\Security","Security",REG_BINARY,01,00,14,80,b8,00,00,00,c4,00,00,00,14,00,00,00,30,00,00,00,02,00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,00,00,02,00,88,00,06,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,04,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,06,00,00,00,00,14,00,00,01,00,00,01,01,00,00,00,00,00,05,0b,00,00,00,00,0



```
0,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
```

```
HKLM,"SYSTEM\ControlSet001\Services\IBSMUTIL\Enum","0",REG_SZ,"ROOT\SYSTEM\0003"
```

```
HKLM,"SYSTEM\ControlSet001\Services\IBSMUTIL\Enum","Count",REG_DWORD,00000001
```

```
HKLM,"SYSTEM\ControlSet001\Services\IBSMUTIL\Enum","NextInstance",REG_DWORD,00000001
```

```
;  
;  
;
```