

PassMark® White Paper

Self booting BurnInTest:

Using BurnInTest on a system with no
Operating System - WinPE 3, 4, 5, 10 and
Windows 8.1 To Go

CD/DVD, USB or PXE network boot



Edition: 4.0
Date: 17 September 2020
BurnInTest Version: 9.1

BurnInTest is a trademark of PassMark software

Contents

Overview..... 3

Audience..... 3

Using WinPE Builder 4

Standard Environment 4

Limitations 4

 WinPE limitations: 4

 Windows To Go limitations: 5

Downloads 5

Building a WinPE 10.0 Environment..... 6

Older Versions 11

Building a WinPE 3.0 Environment..... 11

 Example 64-bit WinPE 3.0 build 18

Adding drivers to a WinPE image..... 19

Deploy an Image using PXE..... 20

Walkthrough of WAIK 64-bit WinPE 24

Building a WinPE 4 environment..... 31

Building a WinPE 5 environment..... 32

Windows 8.1 To Go (WTG) 35

Overview

Many BurnInTest users can benefit from testing PC hardware when there is no Operating System installed, or the Operating System is inoperable. This can be useful for testing PC hardware:

1. In a production line environment,
2. That is to be shipped with Linux,
3. In a known virus free environment and
4. To try to determine the cause of corruption of an Operating System.

This document aims to assist people in setting up an environment that allows PassMark BurnInTest to be used in these situations.

This paper is applicable to the Windows Professional edition of BurnInTest v9.1. This version of BurnInTest requires Microsoft Windows in order to run. To run BurnInTest on a system without an operating system you need to set up:

- 1) A "Pre-install environment" that allows Microsoft Windows to be booted from a CD/DVD, USB Flash Drive or via PXE (Preboot eXecution Environment).
- 2) A portable windows environment – Windows To Go on USB.

This document describes setting up a Microsoft Windows Pre-install environment (WinPE) environment and "Windows To Go" that includes both Windows and BurnInTest V9.1 on a bootable CD/DVD or bootable USB Flash Drive (UFD). The document also describes how to inject new device drivers into the Windows image for updated hardware.

The choice of WinPE or Windows To Go will depend on your Windows licensing, which BurnInTest tests are required and potential issues with offline local disks with Windows To Go.

The choice of which version of WinPE to use will be based on the version of Windows you plan to test on. We would recommend the latest build:

- WinPE 3.0: Built from Windows 7 code base.
- WinPE 3.1: Built from Windows 7 SP1 code base.
- WinPE 4.0: Built from Windows 8 code base. Included in Windows ADK (Windows Kits 8.0).
- WinPE 5.0: Built from Windows 8.1 code base. Included in Windows ADK (Windows Kits 8.1).
- WinPE 5.1: Windows ADK (Windows Kits 8.1 Update) includes 5.0.
- Windows 8.1 To Go: Windows 8.1.
- WinPE 10: Built from Windows 10 code base. Included in Windows ADK for Windows 10.

This document does not intend to cover product licensing issues and it is up to the reader to review this. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Audience

This paper is targeted at companies and individuals that need to run BurnInTest on a system without a working or installed Operating system. It is aimed at people with technical PC knowledge.

Using WinPE Builder

PassMark Software also provides a tool called [WinPE Builder](#) that can simplify this process and provides a way of creating a WinPE environment to suit BurnInTest. For most cases this should be a suitable way of creating a WinPE install, however if heavy customization of the environment is required then you can follow the steps below to manually create a WinPE environment.

Standard Environment

The standard environment described in this document is:

- **WinPE 3.0 to WinPE 10, or Windows 8.1 To Go**
- BurnInTest PRO v9.1 (or higher).
- Hardware including at least 512MB of RAM (32-bit WinPE) or 1GB (64-bit WinPE). Windows 7 certified hardware or higher (Windows 8.1 To Go).

Limitations

WinPE limitations:

Currently, we plan to support the following BurnInTest tests under WinPE:

- CPU
- RAM
- Disk (disk volumes only)
- Optical disk (read tests)
- Network (Standard)
- USB (using the Passmark USB 2.0/3.0 Loopback plugs)
- Parallel Port
- Serial port
- Plugin (BurnInTest plugins)

Currently we do not plan to support the following tests under WinPE:

- Video Playback;
- Sound;
- 2D;
- 3D;
- Printer;
- Tape.

Please note that we have intentionally not restricted any of the BurnInTest tests in software, as I am sure there are WinPE users that have or will get these tests working under WinPE. Send us an email.

Some system information normally displayed by BurnInTest will not be available under WinPE, this includes (but may not be limited to) the Graphics card, HDD model and USB controller/device information.

This guide also does not enable networking, and uses the WINPESH.LINI method to launch the program.

Windows PE 10.0 can be obtained with the Microsoft Windows Assessment and Deployment Kit (ADK).

It will demonstrate an x64 version, and you need to use drivers suitable to your build. It is recommended that you track which drivers you end up putting into your PE. You should only need to install drivers for NIC and non-standard Mass Storage Controllers, such as RAID cards. This guide was prepared and tested on a Windows 10 Pro system.

Note that in our testing the 32-bit version of BurnInTest will not run in a 64-bit WinPE environment – use the 64-bit version of BurnInTest in this scenario.

You will need to run the tools with (elevated) Administrator privileges.

Windows To Go limitations:

Currently, we plan to support all of the BurnInTest tests under Windows To Go.

The main issue for BurnInTest on Windows To Go is that the local disks are placed offline. See the section "[Windows 8.1 To Go \(WTG\)](#)" for more information.

Downloads

Microsoft Windows Assessment and Deployment Kit (ADK) can be downloaded here:

<https://docs.microsoft.com/en-us/windows-hardware/get-started/adk-install>

The Windows PE add-on for the ADK is required as well:

<https://go.microsoft.com/fwlink/?linkid=2120253>

Microsoft devcon utility (for installing serial port and parallel port device drivers in WinPE):

<http://support.microsoft.com/kb/311272>

The latest version of the BurnInTest PRO can be downloaded here:

http://passmark.com/download/bit_download.htm

Building a WinPE 10.0 Environment

This section describes how to build a WinPE 10.0 boot USB with BurnInTest V9.1.

This is a final walkthrough to create a functional PE image that will automatically load BIT upon opening. You need to install the ADK Tools first (as well as BIT) before you should start. Also make sure to have your drivers ready. All commands are done by using the ADK "*Deployment and Imaging Tools Environment*", which is a special paths CMD that will appear in the Start menu after you install that tool.

You will need to launch "*Deployment and Imaging Tools Environment*" with elevated administrator privileges. i.e. Start ... All Programs ... Windows Kits ... Deployment and Imaging Tools Environment.

Note: The 'amd64' text in the DISM commands below refers to a 64bit WinPE build. If you intend to create a 32bit WinPE build, then you will need to substitute the 'amd64' text with 'x86' in the DISM commands below

1. Create the base PE source into a temporary folder.

The destination folder cannot already exist.

`copy amd64 c:\temp\bitpe`

Copy a base image: [ADK Install DIR]\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\en-us\winpe.wim into: **C:\temp\bitpe\Sources** (create this new "**Sources**" folder).

eg:

```
mkdir C:\temp\bitpe\Sources\
```

```
copy "C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\en-us\winpe.wim" C:\temp\bitpe\Sources\
```

2. Mount the WinPE source

```
DISM /Mount-Wim /WimFile:C:\temp\bitpe\sources\winpe.wim /Index:1  
/MountDir:C:\temp\bitpe\mount
```

3. Add some required Windows packages to the image

WMI, Scripting, StorageWMI and HTA packages are required for BurnInTest.

```
DISM /image:c:\temp\bitpe\mount /Add-Package /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OCs\en-us\WinPE-WMI_en-us.cab"
```

```
DISM /image:c:\temp\bitpe\mount /Add-Package /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OCs\en-us\WinPE-HTA_en-us.cab"
```

```
DISM /image:c:\temp\bitpe\mount /Add-Package /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OCs\en-us\WinPE-Scripting_en-us.cab"
```

```
DISM /image:c:\temp\bitpe\mount /Add-Package /PackagePath:"C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OCs\en-us\WinPE-StorageWMI_en-us.cab"
```

4. Install required device drivers for your target Operating System

Place the NIC and Mass Storage driver files that you need into the c:\bitpe\drivers\nic and c:\bitpe\drivers\hddc directory paths and then install.

```
DISM /image:c:\temp\bitpe\mount\ /Add-Driver /driver:c:\temp\bitpe\drivers\nic /recurse /ForceUnsigned
```

```
DISM /image:c:\temp\bitpe\mount\ /Add-Driver /driver:c:\temp\bitpe\drivers\hddc /recurse /ForceUnsigned
```

Note: By including the /recurse switch you tell the command to recurse the drivers' subfolders for valid .inf drivers and by including the /ForceUnsigned you tell the command to ignore driver signing requirements.

To test USB ports with the PassMark USB 2/3 Loopback plug the PMUSB drivers need to be installed in the WinPE build. Copy the driver files (*.inf, *.sys, *.cat etc) to a temporary directory. For example, for a 64-bit WinPE build, copy the 64-bit Passmark USB Loopback driver files to c:\bitpe\drivers\usb and install the INF packages.

```
DISM /image:c:\temp\bitpe\mount\ /Add-Driver /driver:c:\temp\bitpe\drivers\usb /recurse /ForceUnsigned
```

Note: After installing the drivers, you can use the "DISM /Get-Drivers" command to see which drivers are installed.

For example,

```
DISM /image:c:\temp\bitpe\mount\ /Get-Drivers -- View 3rd party drivers  
DISM /image:c:\temp\bitpe\mount\ /Get-Drivers /all -- View all drivers
```

5. Install the BurnInTest software and msvfw32.dll.

After step 2, you can find the directory structure of the PE with Windows Explorer.

First, put the [msvfw32.dll](#) (Microsoft Video for Windows, required regardless of whether you use the Video playback test) into [c:\temp\bitpe\mount\windows\system32](#). This must be the version suitable for your target Operating System. If you use the wrong dll (e.g. 64-bit on 32-bit Windows), BurnInTest will simply not start – and no error message will be displayed.

During the creation of this document the following version of msvfw32.dll was used:
64-bit: 10.0.18362.1, 140KB, from a 64-bit Windows 10 installation (\windows\system32).

Second, in BurnInTest, choose the option to Install to USB (File->Install BurnInTest to a USB drive...). Specify the Installation location Directory as a temporary location (eg c:\temp\bitUSB). You cannot copy directly to the mounted WinPE image as you will get an error.

Specify the installation type as Licensed and paste your license key into the Username and key section.

After the install to the temp location is complete copy the contents of `c:\temp\bitUSB` to `c:\temp\bitpe\mount\program files\BurnInTest`.

If you have a BurnInTest configuration you want to use, place this configuration in the new BurnInTest folder as well. The steps following assume that a configuration, `test.bitcfg`, has been prepared to test the CPU, Optical Drive, RAM, Network and HDD, and has been copied to the `c:\temp\bitpe\mount\program files\BurnInTest` directory.

The following example uses a program called `setres` to set the resolution to 1024x768. The default resolution for Win PE is 800x600. These files have been installed, but they are optional.

Note: If you want your own image for the WinPE background, replace the default image `c:\temp\bitpe\mount\Windows\System32\winpe.bmp` with you own image.

6. Automate the launching of BIT.

There are two methods that you can use to launch BIT. You can either edit `c:\temp\bitpe\mount\windows\system32\startnet.cmd` or create a `winpeshl.ini` file and place it in the `c:\temp\bitpe\mount\windows\system32` directory. For testing purposes, it is recommended you use the `startnet.cmd` method, because you will have access to the command prompt. If you use `winpeshl.ini`, you will not be able to use a command prompt, but will stop regular users from having direct access into the PE itself once booted. You should not use both options, if `winpeshl.ini` is present, it will ignore the `startnet.cmd` file. If you are going to test the hard disk, it will need to be formatted first. **This guide uses the idea that the hard drive is blank, and we will use the PE to format the disk first.** You would not want to use this option if you are going to test computers with Windows or other OS installed on them already.

You may need to create a copy of the `startnet.cmd` file and edit it in another directory as the mounted image imposes restrictions on opening and editing files.

See the full command lines under `startnet.cmd` to see what the batch files are doing.

winpeshl.ini

[LaunchApps]

```
%SYSTEMDRIVE%\Windows\System32\wpeinit.exe
```

```
; If your hard disk is formatted, do not use this line
```

```
%SYSTEMDRIVE%\Windows\System32\diskpart.exe, " /s
```

```
%SYSTEMDRIVE%\Windows\System32\diskpart.txt"
```

```
; Launch BurnInTest
```

```
"%SYSTEMDRIVE%\Program Files\BurnInTest\bit.exe", "-h -x -r -c test.bitcfg"
```

Note the use of quotations – it is important.

Note:

-C [configfilename]

Loads the configuration file specified by [configfilename]

`test.bitcfg` specifies an optional configuration file copied in step 5.

-h

Set the screen resolution to 1024 x 768 with 32-bit color on startup.

-r

Executes the tests immediately without needing to press the go button. It also skips the pre-test warning message.

-x

Skip the DirectX version checks at startup time. This can be useful for users that do not want to install the latest version of DirectX and do not want to use the DirectX tests (eg. 3D tests).

Warning!

Diskpart.exe is only required if your disk is unformatted – this will delete any data on physical drive 0.

If your hard disk is formatted, do not include the disk partitioning tool line shown above.

I.e. The following line should only be used to format a hard disk – it is optional:

```
%SYSTEMDRIVE%\Windows\System32\diskpart.exe,  
"%SYSTEMDRIVE%\Windows\System32\diskpart.txt"
```

Note: BurnInTest 9.1 should automatically configure the [serial ports](#) to allow them to be detected and tested, so no additional configuration should be required to test serial ports.

If you want to [test parallel ports](#) you need to ensure the parallel port service is started and install the drivers. For example:

winpeshl.ini

```
[LaunchApps]
```

```
%SYSTEMDRIVE%\Windows\System32\wpeinit.exe
```

```
"%SYSTEMDRIVE%\Windows\System32\net", "start parport"
```

```
"%SYSTEMDRIVE%\Windows\devcon.exe", "install x:\windows\inf\msports.inf ACPI\PNP0401"
```

```
"%SYSTEMDRIVE%\Windows\devcon.exe", "restart *PNP0401"
```

```
"%SYSTEMDRIVE%\Program Files\BurnInTest\bit.exe", "-h -x"
```

*devcon will need to be downloaded from Microsoft and included in the WinPE build in the Windows directory.

Startnet.cmd

Wpeinit

::If your hard disk is formatted, do not use the following line.

::You will need to specify a diskpart.txt file – example below

```
diskpart /s diskpart.txt
```

```
"x:\Program Files\burnintest\bit.exe" -h -x -r -c test.bitcfg
```

Notes:

BurnInTest 9.1 automatically configures the serial ports to allow them to be detected and tested, so no additional configuration is required to test serial ports.

if you want to test parallel ports you need to ensure the parallel port service is started and install the drivers. Add these commands to Startnet.cmd before bit.exe is started, for example, startnet.cmd:

```
wpeinit
net start parport
devcon install x:\windows\inf\msports.inf ACPI\PNP0401
devcon restart *PNP0401
"x:\Program Files\burnintest\bit.exe" -h -x
```

*devcon will need to be downloaded from Microsoft and included in the WinPE build in the Windows directory.

diskpart.txt file contents:

```
select disk 0
clean
create partition primary
select partition 1
active
format quick fs=ntfs
assign letter=c
```

Warning!

This is only if your disk is unformatted – this will delete any data on physical drive 0.

7. Commit the changes and save the image

Create a winpe.wim image from the local directory.

```
DISM /unmount-Wim /MountDir:c:\temp\bitpe\mount /Commit
```

8. Make the boot disk

To make a bootable USB Flash Drive (UFD):

From the command prompt, partition and format the UFD. Make sure you select the correct disk number, as this will delete everything on the disk (the below example shows a UFD with physical disk number 2 and volume letter X).

```
C:\Users\Administrator>diskpart
DISKPART> list disk
DISKPART> select disk 2
```

SOFTWARE

```
DISKPART> clean
DISKPART> create partition primary
DISKPART> select partition 1
DISKPART> active
DISKPART> format fs=fat32 (or ntfs)
DISKPART> assign
DISKPART> exit
```

NOTE: Some UEFI implementations may not support booting from NTFS so FAT32 is generally recommended

Make sure drive is bootable using bootsect.exe;
bootsect.exe /nt60 x: /force /mbr

Copy required files across;

```
xcopy "c:\temp\bitpe\media\*.*" "x:\" /s /e /f
xcopy "c:\temp\bitpe\sources\winpe.wim" "x:\sources\boot.wim" /s /e /f
```

The USB can now be booted from.

Older Versions

The following sections were written for older versions of WinPE and BurnInTest and have been left here for reference.

Building a WinPE 3.0 Environment

This section describes how to build a WinPE 3.0 boot CD or DVD with BurnInTest V8.0.

This is a final walkthrough to create a functional PE image that will automatically load BIT upon opening. You need to install the WAIK/OPK Tools first (as well as BIT) before you should start. Also make sure to have your drivers ready. All commands are done by using the WAIK or OPK "*Deployment tools Command Prompt*", which is a special paths CMD that will appear in the Start menu after you install that tool.

If you are running Windows 7, you will need to launch "*Deployment tools Command Prompt*" with elevated administrator privileges.
i.e. Start ... All Programs ... Microsoft Windows AIK ... Deployment Tools Command Prompt (Right mouse click and select 'Run as administrator').

Note: The 'x86' text in the DISM commands below refers to a 32bit WinPE build. If you intend to create a 64bit WinPE build, then you will need to substitute the 'x86' text with 'amd64' in the DISM commands below (also refer to "**Example WinPE build**" later in this document).

1. Create the base PE source.

The destination folder cannot already exist.

```
copye x86 c:\bitpe
```

2. Mount the WinPE source

Extract the base image winpe.wim to a local directory:

```
DISM /Mount-wim /WimFile:c:\bitpe\winpe.wim /index:1 /MountDir:c:\bitpe\mount
```

Note: After mounting the base image, you can use the "DISM /Get-Packages" command to see which packages are installed and available for installation.

For example, `DISM /image:c:\bitpe\mount\ /Get-Packages`

3. Install the packages that are needed.

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

```
DISM /image:c:\bitpe\mount /Add-Package /PackagePath:"C:\Program Files\Windows AIK\Tools\PETools\x86\WinPE_FPs\winpe-hta.cab"
```

```
DISM /image:c:\bitpe\mount /Add-Package /PackagePath:"C:\Program Files\Windows AIK\Tools\PETools\x86\WinPE_FPs\winpe-scripting.cab"
```

The packages available are as listed in the Microsoft WinPE documentation. An extract follows:

Package Name	Description
WinPE-FONTSupport- <region>	Additional font support for the following languages: ja-JP, ko-KR, zh-CN, zh-HK, and zh-TW.
WinPE-HTA	HTML Application support. Enables the creation of GUI applications using the Windows Internet Explorer® script engine and HTML services.
Winpe-LegacySetup	The Media Setup feature package. All Setup files from the \Sources folder on the Windows media. Add this package when servicing Setup or the \Sources folder on the Windows media. Must be added with the Setup feature package. To add a new Boot.wim to the media, add either child package in addition to the Setup and Media packages. This package is required to support Windows Server® 2008 R2 installation.
WinPE-MDAC	Microsoft® Data Access Component support. Enables queries to SQL servers with Active Directory Objects (ADO). Example usage: building a dynamic Unattend from unique system information.
WinPE-PPPoE	Enables Point-to-Point Protocol over Ethernet (PPPoE) support. Create, connect, disconnect and delete PPPoE connections from Windows PE.
WinPE-Scripting	Windows Script Host (WSH) support. Enables batch file processing using WSH script objects.
WinPE-Setup	The Setup feature package (parent). All Setup files from the \Sources folder common to Client and Server.
WinPE-Setup-Client	The Client Setup feature package (child). The Client branding files for Setup. Must be added after the Setup feature package.
WinPE-Setup-Server	The Server Setup feature package (child). The Server branding files for Setup. Must be added after the Setup feature package.
WinPE-SRT	The Windows Recovery Environment feature package. Provides a recovery platform for automatic system diagnosis and repair and the creation of custom recovery solutions.
WinPE-WMI	Windows Management Instrumentation (WMI) support. A subset of the WMI providers that enables minimal system diagnostics.
WinPE-WDS-Tools	The Windows Deployment Services tools feature package. Includes APIs to enable a multicast scenario with a custom Windows Deployment Services client and Image Capture utility.

4. Install the required device drivers

Place the NIC and Mass Storage driver files that you need into the c:\bitpe\drivers\nic and c:\bitpe\drivers\hddc directory paths and then install.

```
DISM /image:c:\bitpe\mount\ /Add-Driver /driver:c:\bitpe\drivers\nic /recurse /ForceUnsigned
```

```
DISM /image:c:\bitpe\mount\ /Add-Driver /driver:c:\bitpe\drivers\hddc /recurse /ForceUnsigned
```

Note: By including the /recurse switch you tell to the command to recurse the drivers' subfolders for valid .inf drivers and by including the /ForceUnsigned you tell the command to ignore driver signing requirements.

To test USB ports with the PassMark USB 2.0 Loopback plug the PMUSB drivers need to be installed in the WinPE build. Copy the driver files (*.inf, *.sys, *.cat etc) to a temporary directory. For example,

USING BURNINTEST ON A SYSTEM WITH NO OPERATING SYSTEM - WINPE

SOFTWARE

for a 32-bit WinPE build, copy the 32-bit Passmark USB 2.0 Loopback driver files to `c:\bitpe\drivers\usb` and install the INF package.

```
DISM /image:c:\bitpe\mount\ /Add-Driver /driver:c:\bitpe\drivers\usb /recurse /ForceUnsigned
```

Note: After installing the drivers, you can use the "DISM /Get-Drivers" command to see which drivers are installed.

For example,

```
DISM /image:c:\bitpe\mount\ /Get-Drivers -- View 3rd party drivers
```

```
DISM /image:c:\bitpe\mount\ /Get-Drivers /all -- View all drivers
```

5. Install the BurnInTest software and msvfw32.dll.

After step 2, you can find the directory structure of the PE with Windows Explorer.

First, put the [msvfw32.dll](#) (Microsoft Video for Windows, required regardless of whether you use the Video playback test) into `mount\windows\system32`. This must be the version suitable for your target Operating System. If you use the wrong dll (e.g. 64-bit on 32-bit Windows), BurnInTest will simply not start – and no error message will be displayed.

During the creation of this document the following versions of msvfw32.dll were used:
32-bit: 6.1.7600.16385, 117KB, from a 32-bit Windows 7 installation (`\windows\system32`).
64-bit: 6.1.7600.16385, 141KB, from a 64-bit Windows 7 installation (`\windows\system32`).

Second, in BurnInTest, choose the option to Install to USB (File->Install BurnInTest to a USB drive...). Specify the Installation location Directory as `mount\program files\BurnInTest`, specify the installation type as Licensed and paste your license key into the Username and key section.

If you have a BurnInTest configuration you want to use, place this configuration in the new BurnInTest folder as well. The steps following assume that a configuration, `test.bitcfg`, has been prepared to test the CPU, Optical Drive, RAM, Network and HDD, and has been copied to the `mount\program files\BurnInTest` directory.

The following example uses a program called `setres` to set the resolution to 1024x768. The default resolution for Win PE is 800x600. These files have been installed, but they are optional.

Note: If you want your own image for the WinPE background, replace the default image `mount\Windows\System32\winpe.bmp` with you own image.

6. Automate the launching of BIT.

There are two methods that you can use to launch BIT. You can either edit `\mount\windows\system32\startnet.cmd` or create a `winpeshl.ini` file and place it in the `\mount\windows\system32` directory. For testing purposes, it is recommended you use the `startnet.cmd` method, because you will have access to the command prompt. If you use `winpeshl.ini`, you will not be able to use a command prompt, but will stop regular users from having direct access into the PE itself once booted. You should not use both options, if `winpeshl.ini` is present, it will ignore the `startnet.cmd` file. If you are going to test the hard disk, it will need to be formatted first. This guide uses the idea that the hard drive is blank, and we will use the PE to format the disk first. You would not want to use this option if you are going to test computers with Windows or other OS installed on them

already.

See the full command lines under startnet.cmd to see what the batch files are doing.

winpeshl.ini

[LaunchApps]

```
%SYSTEMDRIVE%\Windows\System32\wpeinit.exe
%SYSTEMDRIVE%\Windows\System32\diskpart.exe, " /s
%SYSTEMDRIVE%\Windows\System32\diskpart.txt"
"%SYSTEMDRIVE%\Program Files\BurnInTest\bit.exe", "-h -x -r -c test.bitcfg"
```

Note the use of quotations – it is important.

Note:

-C [configfilename]

Loads the configuration file specified by [configfilename]

test.bitcfg specifies an optional configuration file copied in step 5.

-h

Set the screen resolution to 1024 x 768 with 32-bit color on startup.

-r

Executes the tests immediately without needing to press the go button. It also skips the pre-test warning message.

-x

Skip the DirectX version checks at startup time. This can be useful for users that do not want to install the latest version of DirectX and do not want to use the DirectX tests (eg. 3D tests).

Warning!

Diskpart.exe is only required if your disk is unformatted – this will delete any data on physical drive 0. If your hard disk is formatted, do not include the disk partitioning tool line shown above.

I.e. The following line should only be used to format a hard disk – it is optional:

```
%SYSTEMDRIVE%\Windows\System32\diskpart.exe,
"%SYSTEMDRIVE%\Windows\System32\diskpart.txt"
```

Please note that at the time of writing the BurnInTest Raw disk test is not supported under WinPE.

Note: BurnInTest 8.0 automatically configures the serial ports to allow them to be detected and tested, so no additional configuration is required to test serial ports.

If you want to test parallel ports you need to ensure the parallel port service is started and install the drivers. For example:

winpeshl.ini

[LaunchApps]

```
%SYSTEMDRIVE%\Windows\System32\wpeinit.exe
"%SYSTEMDRIVE%\Windows\System32\net", "start parport"
"%SYSTEMDRIVE%\Windows\devcon.exe", "install x:\windows\inf\msports.inf ACPI\PNP0401"
"%SYSTEMDRIVE%\Windows\devcon.exe", "restart *PNP0401"
"%SYSTEMDRIVE%\Program Files\BurnInTest\bit.exe", "-h -x"
```

*devcon will need to be downloaded from Microsoft and included in the WinPE build in the Windows directory.

SOFTWARE

Startnet.cmd

```
wpeinit  
diskpart /s diskpart.txt  
"x:\Program Files\burnintest\bit.exe" -h -x -r -c test.bitcfg
```

Notes:

- BurnInTest 8.0 automatically configures the serial ports to allow them to be detected and tested, so no additional configuration is required to test serial ports.
- if you want to test parallel ports you need to ensure the parallel port service is started and install the drivers. Add these commands to Startnet.cmd before bit.exe is started, for example, startnet.cmd:

```
wpeinit  
net start parport  
devcon install x:\windows\inf\msports.inf ACPI\PNP0401  
devcon restart *PNP0401  
"x:\Program Files\burnintest\bit.exe" -h -x
```

*devcon will need to be downloaded from Microsoft and included in the WinPE build in the Windows directory.

diskpart.txt

```
select disk 0  
clean  
create partition primary  
select partition 1  
active  
format quick fs=ntfs  
assign letter=c
```

Warning!

This is only if your disk is unformatted – this will delete any data on physical drive 0.

7. Commit the changes and save the image

Create a winpe.wim image from the local directory.
`DISM /unmount-wim /MountDir:c:\bitpe\mount /Commit`

8. Make the boot disk

Now at this point you can *rename the c:\bitpe\winpe.wim to boot.wim and place it in the ISO\sources folder* to burn a CD, USB Flash Drive or you can add it to the Boot Images in Windows Deployment Services.

9a. To make a bootable CD/DVD

To make an iso image for burning to CD:
`OSCDIMG -n -h -bc:c:\bitpe\etfsboot.com c:\bitpe\iso c:\bitpe.iso`

Now burn the iso image (`c:\bitpe.iso`) to the CD/DVD. You can use the CD/DVD burning software that comes with Windows 7 or use third-party software.

9b. To make a bootable USB Flash Drive (UFD)

From the command prompt, partition and format the UFD. Make sure you select the correct disk number, as this will delete everything on the disk (the below example shows a UFD with physical disk number 2 and volume letter X).

```
C:\Users\Administrator>diskpart
DISKPART> list disk
DISKPART> select disk 2
DISKPART> clean
DISKPART> create partition primary
DISKPART> select partition 1
DISKPART> active
DISKPART> format fs=fat32 (or ntfs)
DISKPART> assign
DISKPART> exit
```

```
C:\Users\Administrator>xcopy c:\bitpe\iso\*.* X:\ /s /e /f
```

NOTE: Some UEFI implementations may not support booting from NTFS so FAT32 is generally recommended

Example 64-bit WinPE 3.0 build

A 64-bit build example:

<Open the "Deployment tools Command Prompt" with administrator privileges>

```
copy c:\amd64 c:\bitpe_x64
```

```
DISM /Mount-wim /WimFile:c:\bitpe_x64\winpe.wim /index:1 /MountDir:c:\bitpe_x64\mount
```

```
DISM /image:c:\bitpe_x64\mount /Add-Package /PackagePath:"C:\Program Files\Windows AIK\Tools\PETools\amd64\WinPE_FPs\winpe-wmi.cab"
```

```
DISM /image:c:\bitpe_x64\mount /Add-Package /PackagePath:"C:\Program Files\Windows AIK\Tools\PETools\amd64\WinPE_FPs\winpe-hta.cab"
```

```
DISM /image:c:\bitpe_x64\mount /Add-Package /PackagePath:"C:\Program Files\Windows AIK\Tools\PETools\amd64\WinPE_FPs\winpe-scripting.cab"
```

<Install required device drivers for your target Operating System – refer Step 4> For example, to install the PassMark USB 2.0 Loopback drivers that are in c:\bitpe_x64\drivers\usb:

```
DISM /image:c:\bitpe_x64\mount /Add-Driver /driver:c:\bitpe_x64\drivers\usb /recurse /ForceUnsigned
```

<Install 64-bit BurnInTest and 64-bit msvfw32.dll for your target Operating System – refer step 5>

<Create startnet.cmd or winpeshl.ini – refer step 6>

```
DISM /unmount-Wim /MountDir:c:\bitpe_x64\mount /Commit
```

<Copy winpe.wim to iso\sources\boot.wim – refer step 8>

```
OSCDIMG -n -h -bc:\bitpe_x64\etfsboot.com c:\bitpe_x64\iso c:\bitpe_x64.iso
```

<Burn c:\bitpe_x64.iso to an optical disk or copy it to a Bootable USB Flash Drive, refer step 9>

Please consult the documentation that comes with the WAIK/OPK Tools for further information or configurations.

Adding drivers to a WinPE image

This section describes how to install device drivers into an existing WinPE image for updated hardware.

1. Make a copy of the WinPE image to work with.

For example, if your WinPE image is on a UFD, copy the contents of the UFD to c:\pctk

2. Create a directory to mount the image into.

e.g. c:\pctk\mount (this must be an empty directory).

3. Copy your driver files

Copy your driver files (*.inf, *.sys, *.cat etc) to a temporary directory, e.g. copy the 32-bit Passmark USB 2.0 Loopback drivers to c:\pctk\drivers.

4. Check if your image file (.wim) has more than 1 image in it.

`DISM /Get-Wiminfo /wimfile:c:\bitpe\sources\boot.wim`

```
...
Index : 1
...
```

5. Mount the WinPE image from the .wim file

Extract the base image to a local directory.

`DISM /Mount-wim /WimFile:c:\bitpe\sources\boot.wim /index:1 /MountDir:c:\bitpe\mount`

6. Install an INF package (typically a driver) to a Windows PE image.

`DISM /image:c:\bitpe\mount\ /Add-Driver /driver:c:\bitpe\drivers /recurse /ForceUnsigned`

```
Deployment Image Servicing and Management tool
Version: 6.1.7600.16385
```

```
Image Version: 6.1.7600.16385
```

```
Searching for driver packages to install...
Found 1 driver package(s) to install.
Installing 1 of 1 - c:\pctk\drivers\PMUSB2.inf: The driver package was successfully installed.
The operation completed successfully.
```

7. Save the changes

Create a WinPE image from the local directory.

`DISM /unmount-Wim /MountDir:c:\bitpe\mount /Commit`

8. Create a bootable CD/DVD or UFD.

Follow steps 8 and 9 in the "Building a Preinstall Environment" to create a bootable CD/DVD or UFD.

Notes:

Please note that we can't guarantee that all hardware will work after including the drivers, as the changes required may be more than just adding drivers. For example, just adding graphics card devices drivers won't allow the 3D test to run as it also needs DirectX (as such we don't support all tests under WinPE).

We have successfully tested this driver installation process with our USB 2.0 Loopback plugs.

Deploy an Image using PXE

There are a number of options to Network boot a client system with a Windows image located on a server. We have tested and documented one of the more common methods: using Microsoft Windows Deployment Services (WDS) to host and deploy a boot image containing BurnInTest for Windows.

Creating a Windows image with BurnInTest (.wim file)

To create a Windows image (e.g. boot.wim), the Microsoft WAIK or OPK can be used as described earlier in this document. In this document example we created a boot.wim containing 64-bit Windows 7 (WinPE) with BurnInTest, the USB 2.0/3.0 loopback plug device drivers, and the configuration for the serial and parallel ports, as described in the earlier section "Building a Preinstall Environment".

Alternatively, a full Windows image could be used and deployed. Please refer to the Microsoft WDS documentation for details.

Setting up a PXE boot server (to host and deploy your Windows boot image)

Microsoft Windows Deployment Services (WDS) will host and deploy the boot image. You need a Windows Server installation and we have used Windows Server 2008 R2 in preparing this documentation.

From Server Manager, add the "Windows Deployment Services" role to the server to install WDS. Default options can be selected during the installation wizard.

Once successfully installed, WDS needs to be configured by selecting Roles-> Windows Deployment Services->Servers, and right click on <your server> e.g. WIN-7MLLIDNESSTD.home.network (in the screenshot below), and select Configure server.

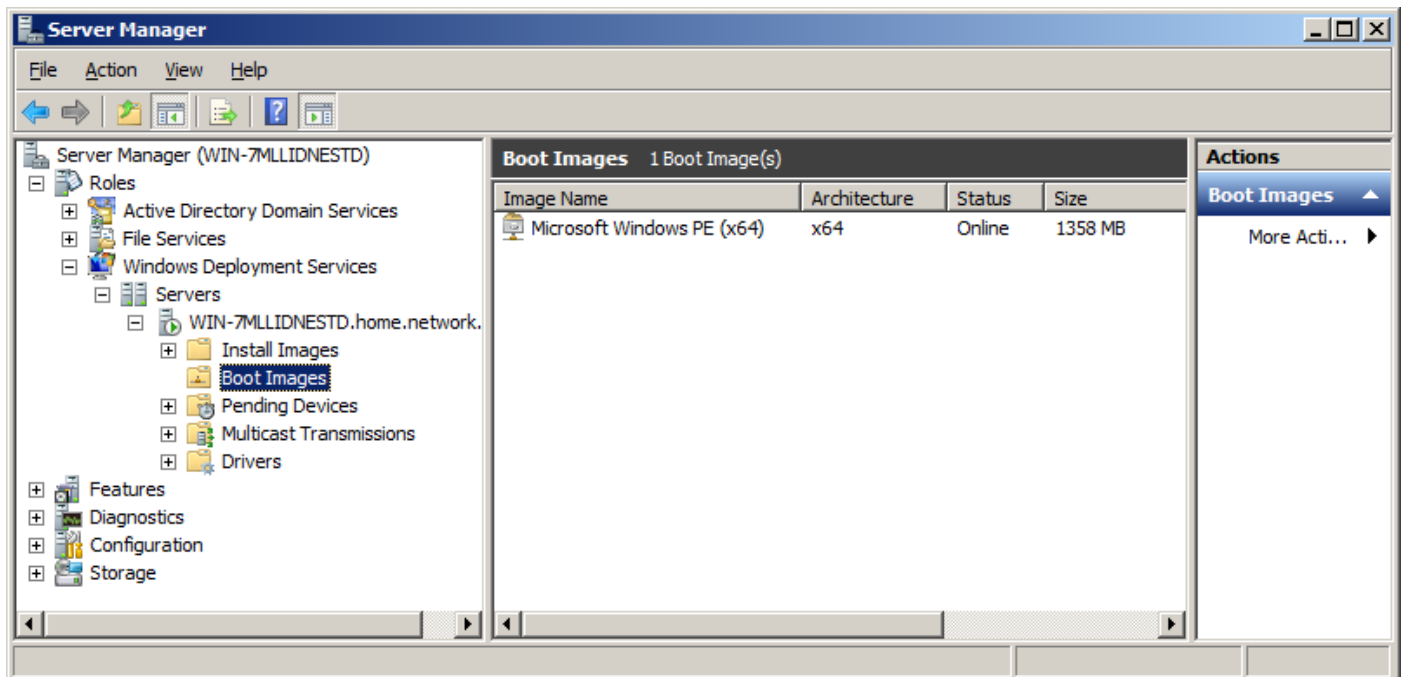
In the configuration options, the defaults can be selected except:

- It is recommended not to use the Windows disk drive as the "Remote installation folder location".
- Set "PXE Server initial settings" to "Respond to all client computers (known and unknown)".
- Uncheck "Add deployment images now".

Add your Windows boot image (e.g. boot.wim): select Roles-> Windows Deployment Services->Servers

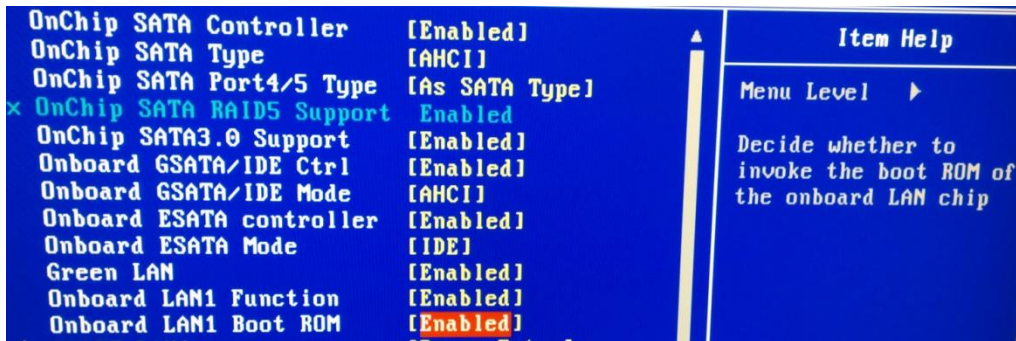
->Servers -> <your server> e.g. WIN-7MLLIDNESSTD.home.network (in the screenshot below), and right click on "Boot images" and select "Add boot image".

Once these tasks have been completed Server Manager should look something like this:

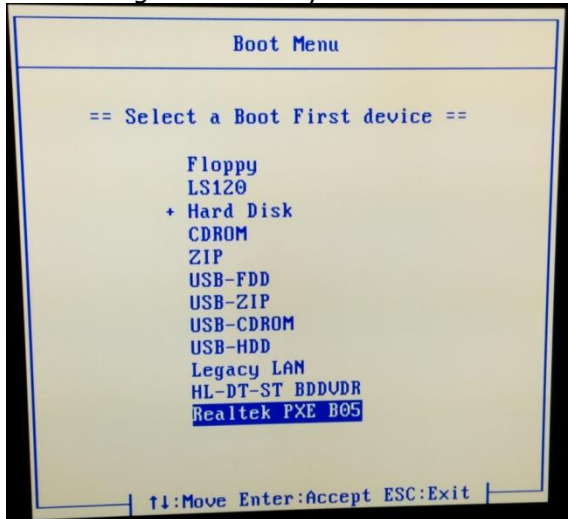


PXE booting

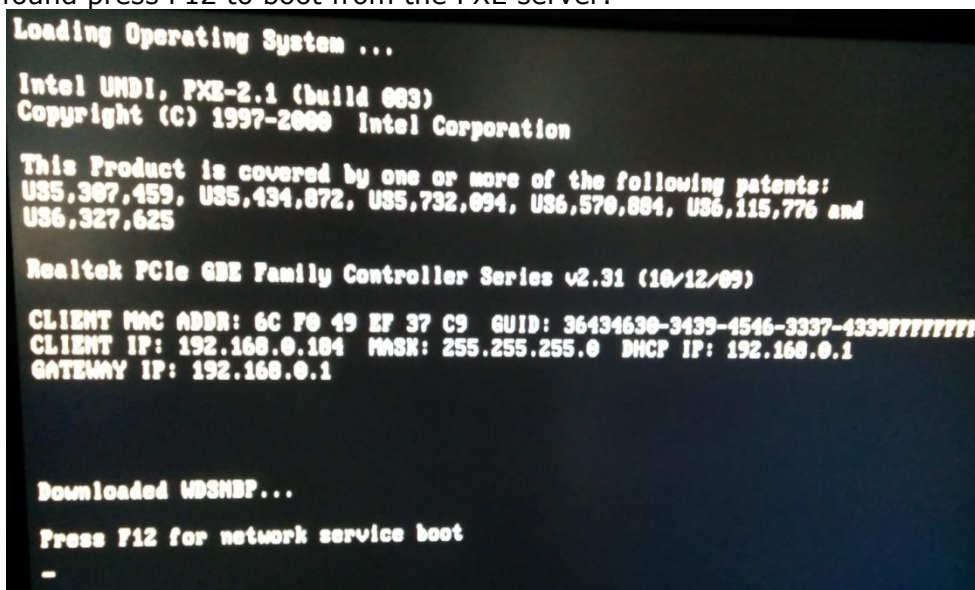
Now to boot this image from the network, you need to enable Network booting in BIOS/UEFI on your client PC(s):



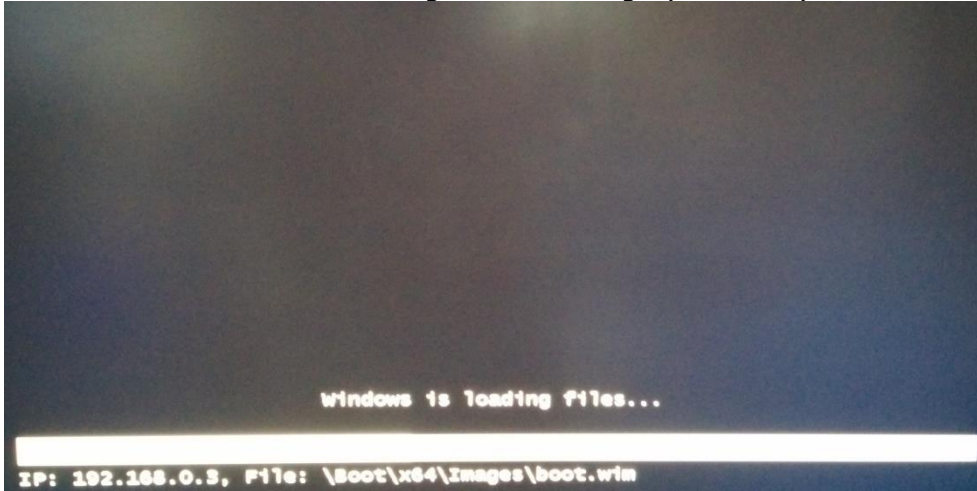
On booting the client system select the PXE boot option, e.g. Realtek PXE B05 in our test case:



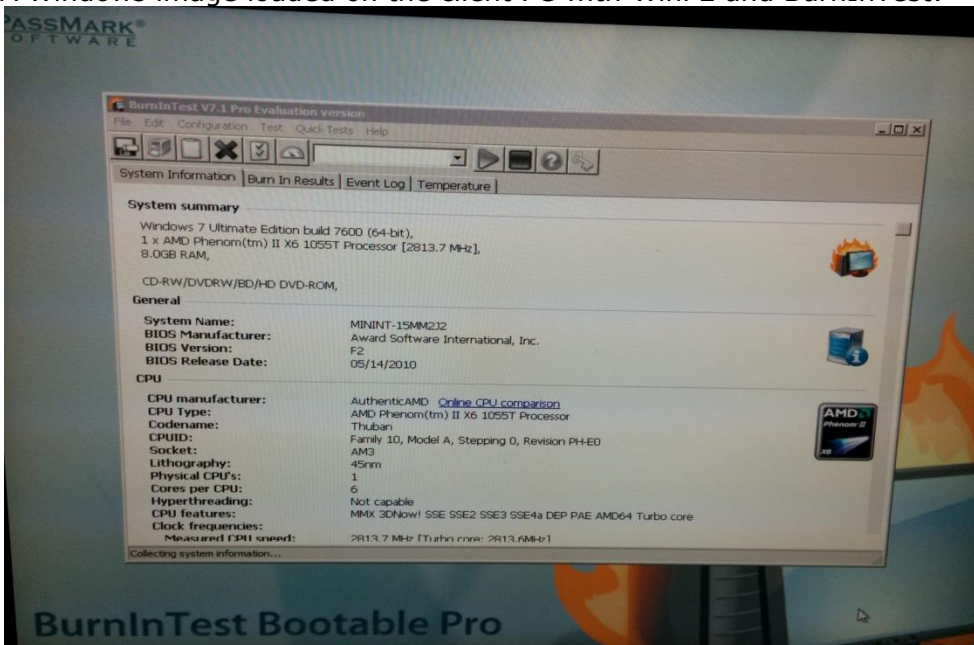
The Network Interface card on the PC will then broadcast to find a PXE server on the network, once found press F12 to boot from the PXE server:



The client PC will now boot using the boot image (boot.wim) on the WDS server (192.168.0.3):



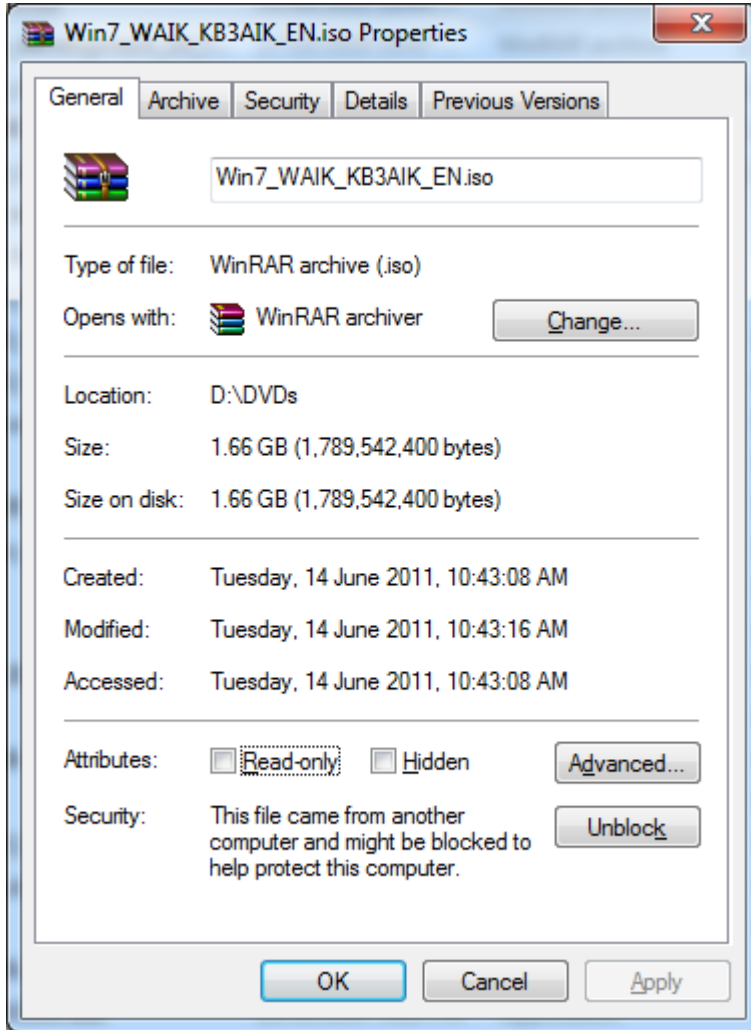
A Windows image loaded on the client PC with WinPE and BurnInTest:



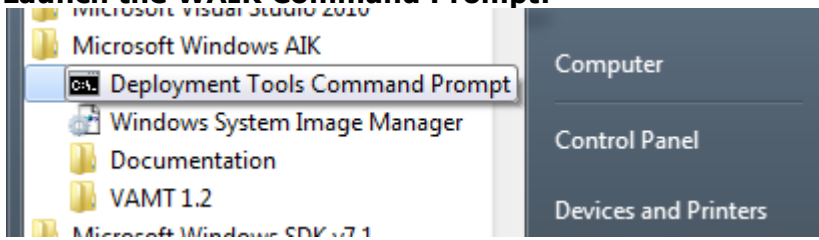
Walkthrough of WAIK 64-bit WinPE

This section shows an example of the build of 64-bit WinPE 3.0 (Windows 7) using the WAIK.

Installing WAIK: This is the WAIK package used in the production of this document. The WAIK needs to be installed once before the first use.



Launch the WAIK Command Prompt:



Command prompt:

```
Administrator: Deployment Tools Command Prompt
Updating path to include dism, oscdimg, imagex
C:\Program Files\Windows AIK\Tools\PETools\
C:\Program Files\Windows AIK\Tools\PETools\..\AMD64;C:\Program Files\Windows
AIK\Tools\PETools\..\x86
C:\Program Files\Windows AIK\Tools\PETools\..\AMD64\Servicing;C:\Program File
s\Windows AIK\Tools\PETools\..\x86\Servicing;

C:\Program Files\Windows AIK\Tools\PETools>copyype amd64 c:\bitpe_x64

=====
Creating Windows PE customization working directory

c:\bitpe_x64
=====

1 file(s) copied.
1 file(s) copied.
1 file(s) copied.
1 file(s) copied.
1 file(s) copied.
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\bcd
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\boot.sdi
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\bootfix.bin
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\efisys.bin
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\efisys_noprompt.bin
```

Walkthrough using command prompt – this shows steps 1,2,3 and 7 from the section “Building a Preinstall Environment”. Please note that the BurnInTest specific steps of 4 and 5 are performed using Windows Explorer (and not the command prompt – hence not shown here).

Updating path to include dism, oscdimg, imagex

```
C:\Program Files\Windows AIK\Tools\PETools\  
C:\Program Files\Windows AIK\Tools\PETools\..\AMD64;C:\Program Files\Windows  
AIK\Tools\PETools\..\x86  
C:\Program Files\Windows AIK\Tools\PETools\..\AMD64\Servicing;C:\Program File  
s\Windows AIK\Tools\PETools\..\x86\Servicing;
```

```
C:\Program Files\Windows AIK\Tools\PETools>copy c:\bitpe_x64
```

```
=====  
Creating Windows PE customization working directory
```

```
c:\bitpe_x64  
=====
```

```
1 file(s) copied.  
1 file(s) copied.  
1 file(s) copied.  
1 file(s) copied.  
1 file(s) copied.  
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\bcd  
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\boot.sdi  
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\bootfix.bin  
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\efisys.bin  
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\efisys_noprompt.bin  
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\etfsboot.com  
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\fonts\chs_boot.ttf  
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\fonts\cht_boot.ttf  
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\fonts\jpn_boot.ttf  
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\fonts\kor_boot.ttf  
C:\Program Files\Windows AIK\Tools\PETools\amd64\boot\fonts\wgl4_boot.ttf  
11 File(s) copied  
C:\Program Files\Windows AIK\Tools\PETools\amd64\EFI\boot\bootx64.efi  
C:\Program Files\Windows AIK\Tools\PETools\amd64\EFI\microsoft\boot\bcd  
C:\Program Files\Windows AIK\Tools\PETools\amd64\EFI\microsoft\boot\fonts\chs_bo  
ot.ttf  
C:\Program Files\Windows AIK\Tools\PETools\amd64\EFI\microsoft\boot\fonts\cht_bo  
ot.ttf  
C:\Program Files\Windows AIK\Tools\PETools\amd64\EFI\microsoft\boot\fonts\jpn_bo  
ot.ttf  
C:\Program Files\Windows AIK\Tools\PETools\amd64\EFI\microsoft\boot\fonts\kor_bo  
ot.ttf  
C:\Program Files\Windows AIK\Tools\PETools\amd64\EFI\microsoft\boot\fonts\wgl4_b  
oot.ttf  
7 File(s) copied  
1 file(s) copied.
```

Success

Updating path to include peimg, cdimage, imagex

```
C:\Program Files\Windows AIK\Tools\PETools\  
C:\Program Files\Windows AIK\Tools\PETools\..\AMD64
```

```
c:\bitpe_x64>DISM /Mount-wim /WimFile:c:\bitpe_x64\winpe.wim /index:1 /MountDir:  
c:\bitpe_x64\mount
```

Deployment Image Servicing and Management tool
Version: 6.1.7600.16385

Mounting image
[=====100.0%=====]
The operation completed successfully.

```
c:\bitpe_x64>DISM /image:c:\bitpe_x64\mount /Add-Package /PackagePath:"C:\Progra  
m Files\Windows AIK\Tools\PETools\amd64\WinPE_FPs\winpe-wmi.cab"
```

Deployment Image Servicing and Management tool
Version: 6.1.7600.16385

Image Version: 6.1.7600.16385

Processing 1 of 1 - Adding package WinPE-WMI-Package~31bf3856ad364e35~amd64~~6.1
.7600.16385
[=====100.0%=====]
The operation completed successfully.

```
c:\bitpe_x64>DISM /image:c:\bitpe_x64\mount /Add-Package /PackagePath:"C:\Progra  
m Files\Windows AIK\Tools\PETools\amd64\WinPE_FPs\winpe-hta.cab"
```

Deployment Image Servicing and Management tool
Version: 6.1.7600.16385

Image Version: 6.1.7600.16385

Processing 1 of 1 - Adding package WinPE-HTA-Package~31bf3856ad364e35~amd64~~6.1
.7600.16385
[=====100.0%=====]
The operation completed successfully.

```
c:\bitpe_x64>DISM /image:c:\bitpe_x64\mount /Add-Package /PackagePath:"C:\Progra  
m Files\Windows AIK\Tools\PETools\amd64\WinPE_FPs\winpe-scripting.cab"
```

Deployment Image Servicing and Management tool
Version: 6.1.7600.16385

Image Version: 6.1.7600.16385

Processing 1 of 1 - Adding package WinPE-Scripting-Package~31bf3856ad364e35~amd6
4~~6.1.7600.16385
[=====100.0%=====]

USING BURNINTEST ON A SYSTEM WITH NO OPERATING SYSTEM - WINPE

SOFTWARE

The operation completed successfully.

```
c:\bitpe_x64>DISM /unmount-Wim /MountDir:c:\bitpe_x64\mount /Commit
```

Deployment Image Servicing and Management tool
Version: 6.1.7600.16385

Image File : c:\bitpe_x64\winpe.wim

Image Index : 1

Saving image

[=====100.0%=====]

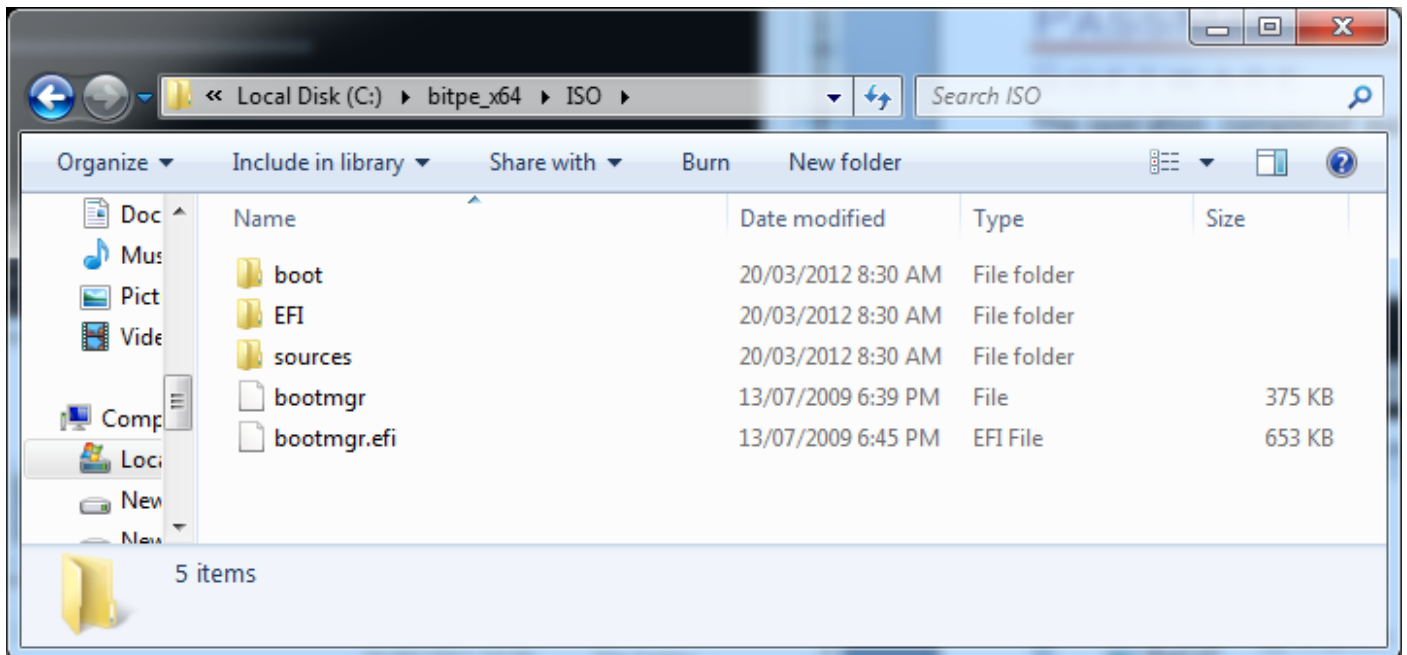
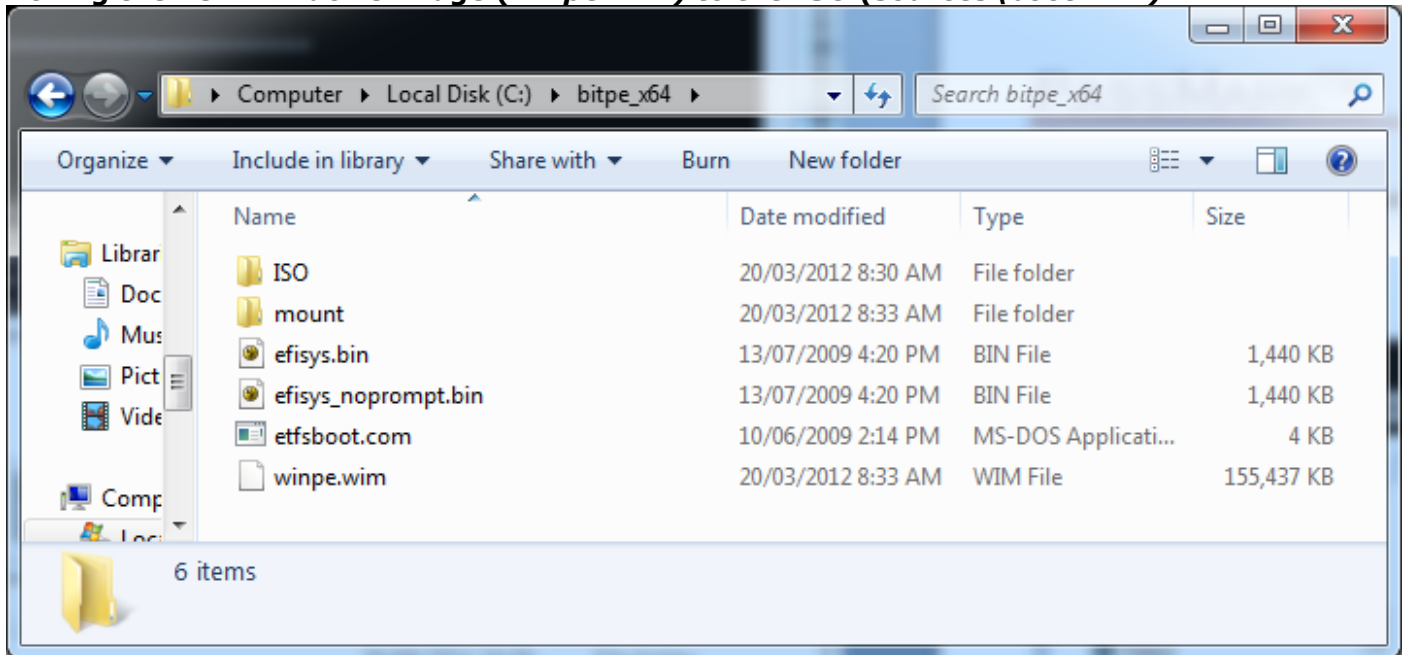
Unmounting image

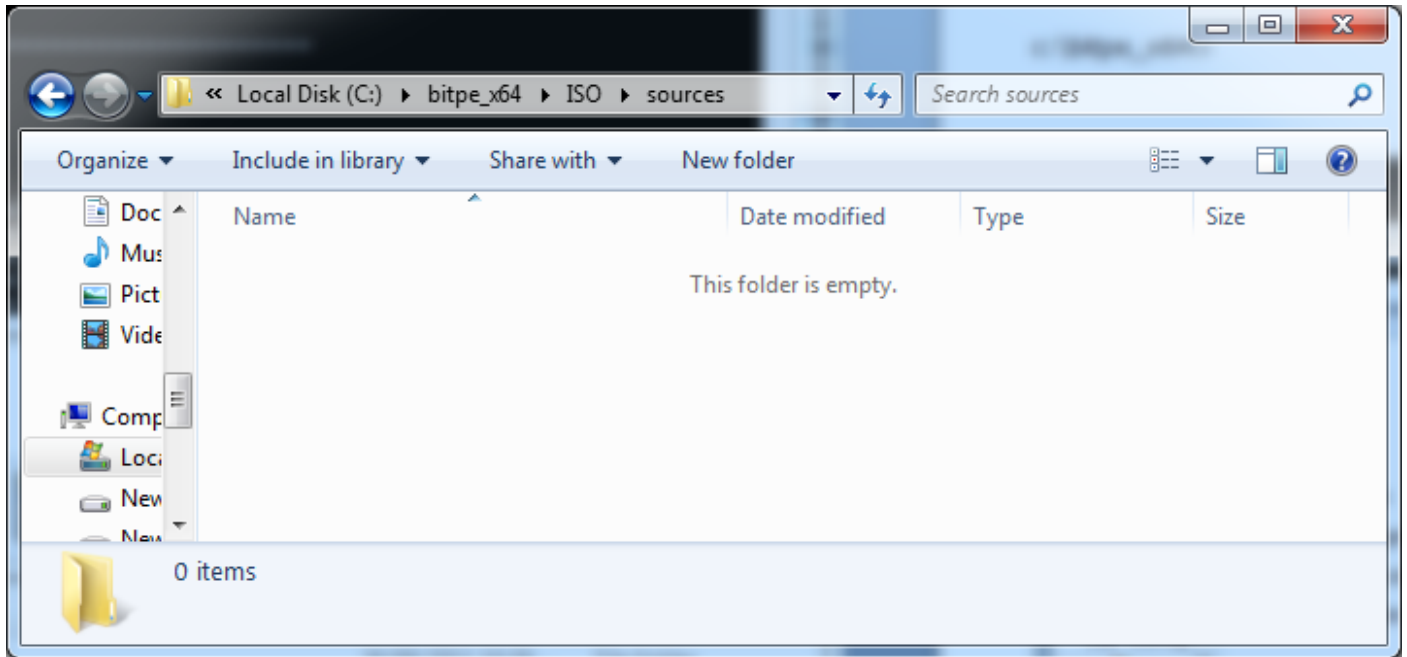
[=====100.0%=====]

The operation completed successfully.

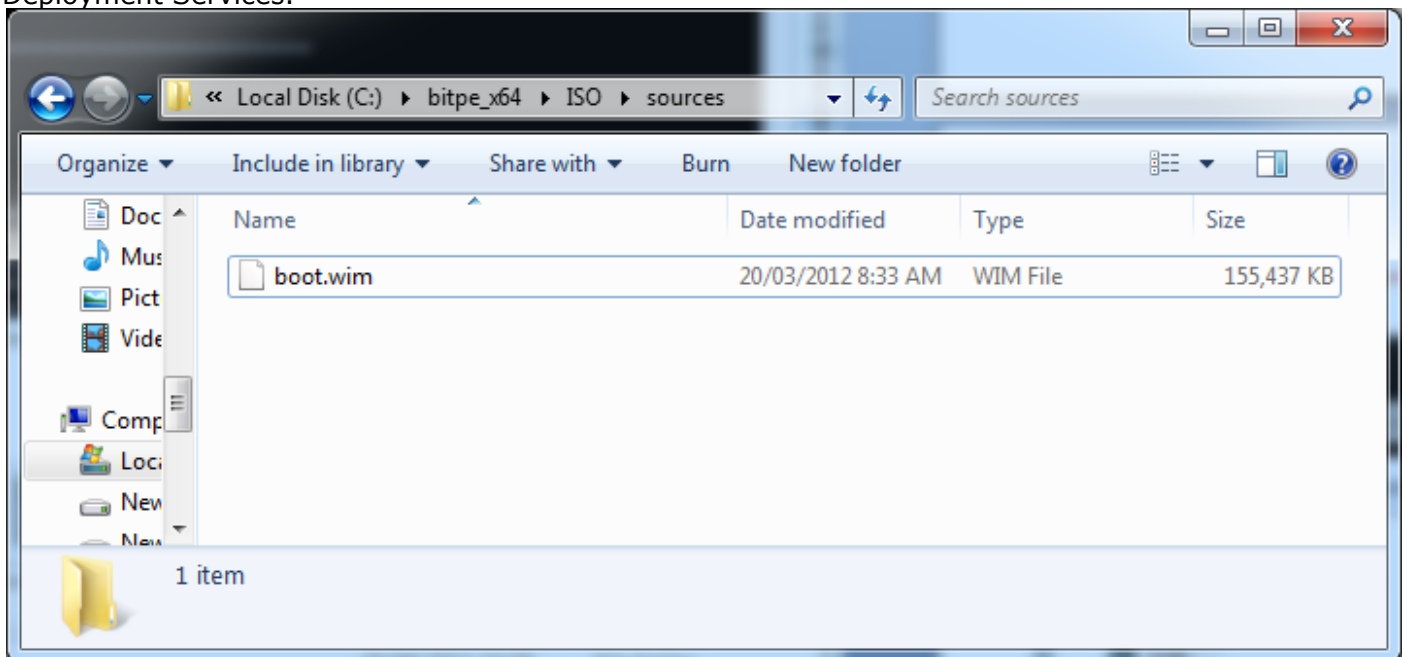
```
c:\bitpe_x64>
```

Moving the new Windows image (*winpe.wim*) to the ISO (*sources\boot.wim*):





Now at this point you can *rename the c:\bitpe_x64\winpe.wim to boot.wim and place it in the ISO\sources folder* to burn a CD, USB Flash Drive or you can add it to the Boot Images in Windows Deployment Services.



Building a WinPE 4 environment

BurnInTest can be installed in a WinPE 4.0 Windows 8 image. 32-bit BurnInTest 8.0 has been tested with 32-bit WinPE 4.

Download

WinPE 4 - Windows ADK (Windows Kits 8.0) is available for download from the Microsoft website. E.g.: <http://www.microsoft.com/download/en/confirmation.aspx?id=28997>

The build process

The build process is almost identical to WinPE 3.0. An example of creating a bootable USB Flash Drive with 32-bit WinPE 4.0 and BurnInTest 8.0 follows:

```
copy c:\bitpe x86 c:\bitpe
```

```
DISM /image:c:\bitpe\mount /Add-Package /PackagePath:"C:\Program Files (x86)\Windows Kits\8.0\Assessment and Deployment Kit\Windows Preinstallation Environment\x86\WinPE_OCs\winpe-wmi.cab"
```

```
DISM /image:c:\bitpe\mount /Add-Package /PackagePath:"C:\Program Files (x86)\Windows Kits\8.0\Assessment and Deployment Kit\Windows Preinstallation Environment\x86\WinPE_OCs\winpe-hta.cab"
```

```
DISM /image:c:\bitpe\mount /Add-Package /PackagePath:"C:\Program Files (x86)\Windows Kits\8.0\Assessment and Deployment Kit\Windows Preinstallation Environment\x86\WinPE_OCs\winpe-scripting.cab"
```

<Install required device drivers for your target Operating System – refer to Step 4 in the section "Building a Preinstall Environment">

<Install 32-bit BurnInTest and 32-bit msfw32.dll for your target Operating System – refer step 5>

<Create startnet.cmd or winpeshl.ini – refer step 6>

```
DISM /unmount-wim /MountDir:c:\bitpe\mount /Commit
```

<Follow Step 9b to make a bootable UFD>

<Copy C:\bitpe\media to the UFD>

Please consult the documentation that comes with the Windows ADK for further information or configurations.

Building a WinPE 5 environment

BurnInTest can be installed in a WinPE 5.0 and WinPE 5.1 Windows 8.1 image. 32-bit BurnInTest 8.0 has been tested with 32-bit WinPE 5.0. 64-bit BurnInTest 8.0 has been tested with 64-bit WinPE 5.1. A PC with Windows 7 (or higher) is required.

Download

WinPE 5.x - Windows ADK (Windows Kits 8.1) is available for download from the Microsoft website. E.g.:

WinPE 5.0: <http://www.microsoft.com/en-us/download/details.aspx?id=30652>

WinPE 5.1: <http://www.microsoft.com/en-au/download/details.aspx?id=39982>

Install the ADK package.

Example of the build process: 32-bit

The build process is almost identical to WinPE 3.0 and 4.0. An example of creating a bootable USB Flash Drive with 32-bit WinPE 5.0 and BurnInTest 8.0 follows:

Open a command window, Start->All Programs->Windows Kits->Windows ADK->Deployment and imaging tools environment.

```
copy x86 c:\bitpe
```

Copy a base image:

Copy *[ADK Install DIR]\Assessment and Deployment Kit\Windows Preinstallation Environment\x86\en-us\winpe.wim* into: **C:\bitpe\Sources** (create this new "**Sources**" folder).

Mount the image:

```
DISM /Mount-Wim /WimFile:C:\bitpe\Sources\boot.wim /Index:1 /MountDir:C:\bitpe\mount
```

Add some Windows packages to the image:

```
DISM /image:c:\bitpe\mount /Add-Package /PackagePath:"C:\Program Files (x86)\Windows Kits\8.1\Assessment and Deployment Kit\Windows Preinstallation Environment\x86\WinPE_OCs\winpe-wmi.cab"
```

```
DISM /image:c:\bitpe\mount /Add-Package /PackagePath:"C:\Program Files (x86)\Windows Kits\8.1\Assessment and Deployment Kit\Windows Preinstallation Environment\x86\WinPE_OCs\winpe-hta.cab"
```

```
DISM /image:c:\bitpe\mount /Add-Package /PackagePath:"C:\Program Files (x86)\Windows Kits\8.1\Assessment and Deployment Kit\Windows Preinstallation Environment\x86\WinPE_OCs\winpe-scripting.cab"
```

<Install required device drivers for your target Operating System – refer to Step 4 in the section "Building a Preinstall Environment"> e.g to install the PassMark USB 2.0 and 3.0 Loopback drivers, copy the 32-bit driver files to c:\bitpe\drivers\usb:

```
DISM /image:c:\bitpe\mount\ /Add-Driver /driver:c:\bitpe\drivers\usb /recurse /ForceUnsigned
```

<Install 32-bit BurnInTest and 32-bit msvfw32.dll for your target Operating System – refer step 5>

<Create startnet.cmd or wineshl.ini – refer step 6>

`DISM /unmount-Wim /MountDir:c:\bitpe\mount /Commit`

<Follow Step 9b to make a bootable UFD>

<Copy C:\bitpe\media to the UFD>

<Copy C:\bitpe\sources\boot.wim to the UFD sources directory>

Please consult the documentation that comes with the Windows ADK for further information or configurations.

Example of the build process: 64-bit

An example of creating a bootable USB Flash Drive with 64-bit WinPE 5.1 and BurnInTest 8.0 follows:

Open a command window, Start->All Programs->Windows Kits->Windows ADK->Deployment and imaging tools environment.

`copy c:\amd64 d:\bitpex64`

Copy a base image:

Copy *[ADK Install DIR]\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\en-us\winpe.wim* into: **C:\bitpe\Sources** (create this new "Sources" folder).

Mount the image:

`DISM /Mount-Wim /WimFile:d:\bitpex64\Sources\boot.wim /Index:1 /MountDir:d:\bitpex64\mount`

Add some Windows packages to the image:

`DISM /image:d:\bitpex64\mount /Add-Package /PackagePath:"C:\Program Files (x86)\Windows Kits\8.1\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\winpe-wmi.cab"`

`DISM /image:d:\bitpex64\mount /Add-Package /PackagePath:"C:\Program Files (x86)\Windows Kits\8.1\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\winpe-hta.cab"`

`DISM /image:d:\bitpex64\mount /Add-Package /PackagePath:"C:\Program Files (x86)\Windows Kits\8.1\Assessment and Deployment Kit\Windows Preinstallation Environment\amd64\WinPE_OC\winpe-scripting.cab"`

<Install required device drivers for your target Operating System – refer to Step 4 in the section "Building a Preinstall Environment"> e.g to install the PassMark USB 2.0 and 3.0 Loopback drivers, copy the 64-bit driver files to c:\bitpe\drivers\usb:

`DISM /image:d:\bitpex64\mount /Add-Driver /driver:d:\bitpex64\drivers\usb /recurse /ForceUnsigned`

<Install 64-bit BurnInTest to D:\bitpex64\mount\Program Files\BurnInTest and 64-bit msvfw32.dll for your target Operating System – refer step 5>

<Create startnet.cmd or winpeshl.ini – refer step 6>

`DISM /unmount-Wim /MountDir:d:\bitpex64\mount /Commit`

<Follow Step 9b to make a bootable UFD>

<Copy C:\bitpe\media to the UFD>

<Copy C:\bitpe\sources\boot.wim to the UFD sources directory>

Please consult the documentation that comes with the Windows ADK for further information or configurations.

Windows 8.1 To Go (WTG)

Windows To Go has many advantages over WinPE. The Windows To Go workspace is a full Windows 8 environment, where-as WinPE is a cut down version of Windows with a number of limitations. Under WTG, all of the BurnInTest tests* are supported, including the DirectX 3D, 2D and video tests, which are not supported under WinPE.

The instructions for making a bootable USB drive using Windows To Go are fairly straight forward and covered by Microsoft here:

<http://social.technet.microsoft.com/wiki/contents/articles/6991.windows-to-go-step-by-step.aspx>

Some points to note:

- You will need Windows 8.1 Enterprise.
- You will need a USB drive of at least 32GB. Microsoft recommends these:
http://technet.microsoft.com/en-us/library/f82d1a0a-d8f7-4e8a-86a6-704166969a42#wtg_hardware
- We used 64-bit Windows 8.1 Enterprise and a Western Digital USB 3.0 512GB hard disk for our testing.
- After installing WTG, you will need to install BurnInTest and any drivers you need (like PassMark USB 2.0 and USB 3.0 Loopback plug drivers) on your WTG drive as per a normal installation.
- If testing disk or optical disk volumes then it is suggested to use the automatic selection options as the volume letters may change between systems.
- Note: From BurnInTest 8.0.1014, if the automatic disk selection option is selected, the WTG USB disk (drive C:) is excluded from the test.

***My local disks are not visible in BurnInTest?**

By default, WTG places all local disks in an offline state to "prevent data leakage" from WTG to the local system. It does this via a SAN policy setting of 4 (normally 1). So, by default, the local physical disks are read only and the disk volumes are not mounted for these offline disks. In this case, you can't use the normal BurnInTest disk volume tests or the physical disk tests that write to the physical disk. You can still run a physical disk read test on all of the local offline drives.

You can place the offline drives online, and hence test them as normal with BurnInTest. Once this is done, then it will apply for subsequent system boots. There are some issues with putting the local drives online: according to the Microsoft documentation:

"It is strongly recommended that you do not mount internal hard drives when booted into the Windows To Go workspace. If the internal drive contains a hibernated Windows 8 operating system, mounting the drive will lead to loss of hibernation state and therefor user state or any unsaved user data when the host operating system is booted. If the internal drive contains a hibernated Windows 7 or earlier operating system, mounting the drive will lead to corruption when the host operating system is booted."

Hence the following assumes you are testing systems that are not hibernated and do not have important customer information on them – e.g. they are pre-release test systems. In saying this, we didn't come across any issues during our testing.

To place a disk online once WTG has booted, either:

- 1) Using Disk Management (Win+X -> Disk Management -> Storage). Right click on a Disk shown as "Offline", and select "Online", this will make the drive writeable and mount disk volumes. Then start BurnInTest and test disks as you normally would.
- 2) Use Diskpart, e.g. to make disk 0 online:
DISKPART.EXE
DISKPART> san policy=OnlineAll
DISKPART> select disk 0

SOFTWARE

```
DISKPART> ATTRIBUTES DISK CLEAR READONLY  
DISKPART> ONLINE DISK
```

This could be automated by creating a batch file to put the disk(s) online, then start BurnInTest, e.g. runbit.bat:

```
diskpart /s script.txt  
bit.exe
```

where script.txt is e.g.:

```
san policy=OnlineAll  
select disk 0  
ATTRIBUTES DISK CLEAR READONLY  
ONLINE DISK
```

To place local internal disks online during boot is possible during the creation of the boot disk. We have not tested this at PassMark. Microsoft have information on configuring the SAN policy here:

[http://technet.microsoft.com/en-us/library/cc749466\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc749466(v=ws.10).aspx)

Additional information

Please see:

<http://blogs.windows.com/business/2013/02/07/answering-your-top-questions-windows-to-go/>