



WinPE Boot.wim

to be used as an internal resource only

Contents

- [Creating a custom winpe boot.wim- SCCM](#)
- [Creating a bootable USB Drive](#)
- [Creating a winpe boot.wim with parameter errors](#)

Creating a custom WinPE Boot.wim

When using System Center Configuration Manager for OS deployment you always use WinPE as your boot media/environment for deploying the actual image...

Normally you can just use the boot image supplied with SCCM (a WinPE 3.0 wim, taken from WAIK and placed in [\\\"SiteServer\"SMS \"SiteCode\"osd\boot\"architecture\"boot.wim](#)), with out any further considerations. Well you might have to inject some NIC and Storage drivers, but that's it!

A problem you might run into though, is lacking functionality of that WinPE build, like no HTA, MDAC/ADO support etc, or you might need to ad a "Custom media hook" to run a script prior to starting the task sequence. If that's the case you will have to make a custom boot image.

Steps

1. Use (WAIK) To create a WinPE 3.0 boot image

- Go to <Start><All Programs><Microsoft Windows AIK> and start **Windows PE Tools Command Prompt**

- Type **COPYPE X86 C:\WinPEx86** (this is for 32bit, for 64bit change x86 to amd64)

2. Mount the boot image to a temporary directory.

- **imagex.exe /mountrw C:\WinPEx86\winpe.wim 1 C:\WinPEx86\mount**

3. Install optional components (Scripting, WMI and XML are required to perform SCCM OS deployments – so they must be installed)

-**peimg.exe /install=*Scripting* C:\WinPEx86\mount\windows** (support for .vbs, .js and .wsf – **mandatory for SCCM**)

-**peimg.exe /install=*WMI* C:\WinPEx86\mount\windows** (support for WMI – **mandatory for SCCM**)

-**peimg.exe /install=*XML* C:\WinPEx86\mount\windows** (support for XML – **mandatory for SCCM**)

_____ optional components _____

-**peimg.exe /install=*HTA* C:\WinPEx86\mount\windows** (support for HTA)



WinPE Boot.wim

to be used as an internal resource only

-peimg.exe /install=*MDAC* C:\WinPE86\mount\windows (support for Database access – mandatory if using MDT DB)

-peimg.exe /install=*Fonts* C:\WinPE86\mount\windows (support for ASIA font types)

4. Copy additional files to the Boot.wim. This step is optional and you could add any number of files and programs.

- Copy **Trace32** to C:\WinPE86\mount\windows\system32 (tool for viewing LOG files – which is a must have)

5. Save a copy of your customised Boot.wim, as the boot.wim used for SCCM will be closed for updates!

- **ImageX /unmount /commit C:\WinPE86\mount**

- **Xcopy C:\WinPE86\winpe.wim C:\WinPE86\source\winpe.wim /y** (this will be the image to use next time you want to customize)

6. Finalize the boot image using the following command:

- **imageX.exe /mountrw C:\WinPE86\winpe.wim 1 C:\WinPE86\mount**

- **peimg.exe /prep C:\WinPE86\mount\windows /f** (this will make the Boot image ready for SCCM)

- **ImageX /unmount /commit C:\WinPE86\mount**

7. Export the custom boot image. This step is necessary to reduce the image footprint (reduce the size)

- **imageX.exe /export C:\WinPE86\winpe.wim 1 C:\WinPE86\boot.wim**

8. Import the boot.wim into Configuration Mgr.

- Right click **Boot Images** and choose **Add Boot Image**. Point to the newly created **Boot.wim** and follow the wizard..

WinPE Boot.wim

to be used as an internal resource only

Creating the WinPE image

A Bootable Windows Preinstallation USB stick can be a useful tool for any PC fix-it guy's toolbox. Here's how to make one and use it for maximum effect.

Not counting download time, it will take about an hour to complete this project. Here's what you must do to create the necessary files to copy to your UFD, presented in the order in which these steps must be completed:

- 1. Download the Windows AIK and ISO Recorder v3, if you need or want it (both are linked to from the previous page).
- 2. Burn the ISO image onto a DVD, then run the file called startcd.exe to launch Windows AIK. Here's what you'll see on-screen:



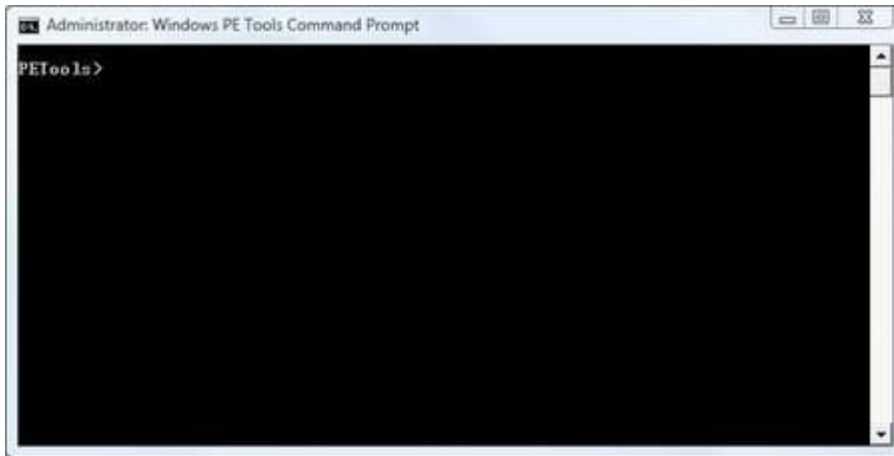
- 3. Click the option labeled "Windows AIK Setup" (Please note that platforms upon which you install AIK must meet this software's Technician PC requirements, which means a Vista PC with a CD/DVD burner, image-burning software, .NET Framework 2.0, and network access. RAM must at least total 512 MB, while 1 GB or more is recommended. See "Build a [Technician Computer](#)" for all the details). Next, click your way through the installation screens to make the WAIK tools available on your desktop.



WinPE Boot.wim

to be used as an internal resource only

- 4. Click Start> All Programs> Windows AIK> then Windows PE Tools Command Prompt. This produces the following screen (we'll show all additional commands in this display frame, where I changed the command prompt to **PETools>**):



- 5. Type `Copyype.cmd x86 C:\winpe_x86` at the command line. Here, use x86 for a 32-bit environment and x64 for a 64-bit environment. The location on your PC where all the various WinPE binaries, optional components, and directories will be created is at `C:\winpe_x86`.

WinPE Boot.wim

to be used as an internal resource only

```

Administrator: Windows PE Tools Command Prompt
PETools> copyype.cmd x86 C:\winpe_x86

=====
Creating Windows PE customization working directory
C:\winpe_x86
=====

1 file(s) copied.
1 file(s) copied.
C:\Program Files\Windows AIK\Tools\PETools\x86\boot\bcd
C:\Program Files\Windows AIK\Tools\PETools\x86\boot\boot.sdi
C:\Program Files\Windows AIK\Tools\PETools\x86\boot\bootfix.bin
C:\Program Files\Windows AIK\Tools\PETools\x86\boot\etfsboot.com
C:\Program Files\Windows AIK\Tools\PETools\x86\boot\fonts\chs_boot.ttf
C:\Program Files\Windows AIK\Tools\PETools\x86\boot\fonts\cht_boot.ttf
C:\Program Files\Windows AIK\Tools\PETools\x86\boot\fonts\jpn_boot.ttf
C:\Program Files\Windows AIK\Tools\PETools\x86\boot\fonts\kor_boot.ttf
C:\Program Files\Windows AIK\Tools\PETools\x86\boot\fonts\wg14_boot.ttf
9 File(s) copied
C:\Program Files\Windows AIK\Tools\PETools\x86\EFI\microsoft\boot\bcd
C:\Program Files\Windows AIK\Tools\PETools\x86\EFI\microsoft\boot\fonts\chs_boot
.ttf
C:\Program Files\Windows AIK\Tools\PETools\x86\EFI\microsoft\boot\fonts\cht_boot
.ttf
C:\Program Files\Windows AIK\Tools\PETools\x86\EFI\microsoft\boot\fonts\jpn_boot
.ttf
C:\Program Files\Windows AIK\Tools\PETools\x86\EFI\microsoft\boot\fonts\kor_boot
.ttf
C:\Program Files\Windows AIK\Tools\PETools\x86\EFI\microsoft\boot\fonts\wg14_boo
t.ttf
6 File(s) copied
1 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\..\x86

PETools>

```

- 6. Next, you can copy any tools or utilities that you might wish to use from the WAIK Tools directory for your architecture (x86 for 32-bit Vista and x64 for 64-bit Vista) into the ISO subdirectory one level below C:\winpe_x86. I usually grab Imagex.exe and the Package Manager using the commands shown in the next screen capture.



WinPE Boot.wim

to be used as an internal resource only

Create WinPE boot.wim with parameter errors

Well after a few hardware failures at work and many hours rebuilding servers that are *supposed* to have RAID 5 on them, I'm back online.

(They had RAID enabled at the BIOS level, and there's a BIOS boot based tool that let's you create/delete stripes etc. BUT you need to use the 'Windows' version of their tool to 'REPAIR' the volume. Now if you can't boot to an O/S....here in lies my problem... I want to plug the new drive in and 'boom' rebuild done... but no!)

I wanted to share a tip which came out of all of this – **Creating a Windows 7 Boot WinPE USB Key.**

You know the 20 odd things you have lying around and are wondering what you could do with 1GB, 512MB etc keys.

You can make a boot disk out of them...cool... (provided your machine supports booting from USB, unlike my mac_mini at home running Windows 7 :)

You could make an x86 boot and on another create a x64 AND you can still store your data.

It's not too tough really – couple of things you need upfront

1. USB Key, drive or whatever
2. [Windows Application Installation Kit for Windows 7 RC](#) (WAIK) – free download with the WinPE 'image' in it and all the tools (a WIM file)
3. Some of your favourite recovery tools to add to your usb key

Ok good to go.

Let's start

1. Install the WAIK on your local machine – it installs folders for x86, amd64 and i64...cool
2. From the Start Menu->All Programs->Microsoft Windows AIK->Deployment Tools Command Prompt
3. Execute the command **copype x86 c:\PEBuild** (replace **x86** for **amd64** if you want 64bit. This will create the directory c:\PEBuild and copy the required files to get started.

```
Administrator: Deployment Tools Command Prompt
Updating path to include dism, oscdimg, imagex

C:\Utils\WAIK\Tools\PETools\
C:\Utils\WAIK\Tools\PETools\..\AMD64;C:\Utils\WAIK\Tools\PETools\..\x86
C:\Utils\WAIK\Tools\PETools\..\AMD64\Serviceing;C:\Utils\WAIK\Tools\PETools\..\
\x86\Serviceing;

C:\Utils\WAIK\Tools\PETools>copype x86 c:\PEBuild
```

4. You'll notice now that you should be in the **C:\PEBuild** folder – this is our 'working space' and there will be a couple of sub-directories here.
5. Let's mount the **Image** so we can manipulate files, and then later save them back into the Image for deployment, as follows:
 1. run the command `imagex /mountrw winpe.wim 1 mount` (this will mount the vanilla winpe.wim and create a 'mount' directory for us to use)

- this didn't work for me on 2 machines that I tried as follows (I was pretty happy)

WinPE Boot.wim

to be used as an internal resource only

```

Administrator: Deployment Tools Command Prompt

C:\PEBuild>imagex /mountrw winpe.wim 1 mount

ImageX Tool for Windows
Copyright (C) Microsoft Corp. All rights reserved.

Mounting: [C:\PEBuild\winpe.wim, 1] -> [C:\PEBuild\mount]...

[ 100% ] Mounting progress

Error mounting image.

The parameter is incorrect.

C:\PEBuild>_
  
```

if this works for you – then great, it’s meant to. (there maybe open files etc)

Fortunately I had a **PLAN B** that works regardless (alot of the documentation use ImageX)

Issue the command from the C:\PEBuild dir: **dism /mount-wim /wimfile:winpe.wim /index:1 /mountdir:c:\PEBuild\Mount**

```

Administrator: Deployment Tools Command Prompt

Directory of C:\PEBuild
05/09/2009  12:02 AM    <DIR>          .
05/09/2009  12:02 AM    <DIR>          ..
10/15/2008  07:35 AM             4,096 etfsboot.com
05/08/2009  10:41 PM    <DIR>          ISO
05/09/2009  12:02 AM    <DIR>          mount
05/09/2009  12:02 AM             136,078,604 winpe.wim
                2 File(s)    136,082,700 bytes
                4 Dir(s)   57,886,724,096 bytes free

C:\PEBuild>dism /mount-wim /wimfile:winpe.wim /index:1 /mountdir:c:\pebuild\mount
t
Deployment Image Servicing and Management tool
Version: 6.1.7000.0

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

C:\PEBuild>
  
```


WinPE Boot.wim

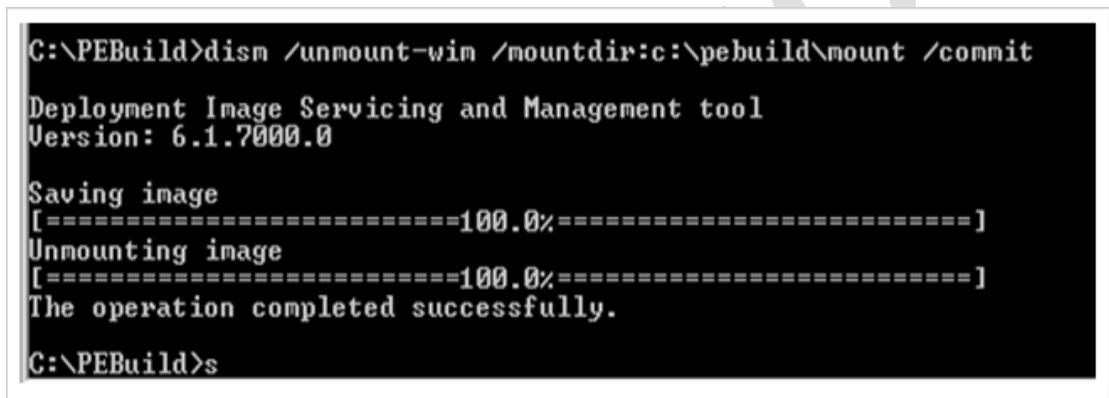
to be used as an internal resource only

2. Go to the **Mount Directory** to see several folders – if you used my PlanB approach you can store the files in Program files if you like, or wherever. If you need access to the **Startup** folder it's hidden in the same dir as Program Files.

Copy your Win32/64 tools and utilities here to the mount dir (once fully booted up with WinPE, you will be in a RamDisk with the drive letter of X: – good idea to always use environment vars to get special folders if you need to)

You can add other support like scripting, powershell etc – check out the packages

3. Once done – **commit the changes to the WinPE.wim** with the command
 - a. ImageX Cmd: **imagex /unmount /commit /c:\pebuild\mount**
 - b. dism Cmd: **dism /unmount-wim /mountdir:c:\pebuild\mount /commit**



```
C:\PEBuild>dism /unmount-wim /mountdir:c:\pebuild\mount /commit
Deployment Image Servicing and Management tool
Version: 6.1.7000.0

Saving image
[=====100.0%=====]
Unmounting image
[=====100.0%=====]
The operation completed successfully.

C:\PEBuild>
```

4. Copy the now *modified* WinPe.Wim image file to the correct directory (and we rename) so it runs smoothly in the boot sequence. Run **copy /y winpe.wim iso/sources/boot.wim**
5. (optional step) To create an ISO of your work, issue the following command:
ocsdimg -n -b"c:\Program Files\Windows AIK\Tools\x86\boot\etfsboot.com" c:\PEBuild\ISO c:\PEBuild\WinPE.iso
6. Next let's add our good work to our USB key and we're done (WE WILL ERASE YOUR KEY – back it up if needed. you can copy your stuff back on when we've finished)
 1. Plug your USB key in and we're going to ntfs format it, create an active primary partition and assign a drive letter to it.
 2. From your command prompt use **diskpart** as follows:
 1. **c:\PEBuild\diskpart**
 2. **list disk** (you should see your USB drive come up in the list – this is the drive # to work with e.g. 1)
 3. **sel disk 1** (double check from the previous command on what disk number your USB is on – mine is 1)
 4. **clean**
 5. **create part primary**
 6. **sel part 1**
 7. **format fs=NTFS QUICK**
 8. **assign**
 9. **exit**
 10. viola! your usb is now prepared.
7. We need to do 2 final steps – copy everything within the **ISO** directory straight to your USB key. e.g. **c:\PEBuild\ISO** (from Explorer you'll be able to get to your USB Key easily through the SendTo right mouse menu)



WinPE Boot.wim

to be used as an internal resource only

Take note of your USB Key's DRIVE letter

1. Write the 'bootsector' as follows: **c:\program files\Windows AIK\Tools\PETools\x86\bootsect /nt60** <drive letter of your USB Key>

ELITE SOLUTIONS