ELITE
ESJ

Step-by-Step Guide for Creating a Window PE Image

to be used as an internal resource only

This walkthrough describes how to create a customized Windows PE image. The primary tool for customizing Windows PE 2.0 is PEImg, a Windows PE command-line tool. After creating a customized image, you can deploy the image to a hard disk or create a bootable Windows PE RAM disk on a CD-ROM, a USB flash drive (UFD) or a hard disk.

Note

A Windows PE RAM disk boots directly into memory and is assigned the drive letter X, which does not correspond to the media (for example, a UFD or a CD-ROM) from which you booted. Ensure that you have sufficient memory to support the size of your Windows PE image plus any additional memory requirements, for example, if you plan on running any customized applications that need additional working memory.

Prerequisites

To complete this walkthrough, you need the following:

- A technician computer that provides all the tools and the source files. For more information, see Building a Technician Computer.

Step 1: Set up a Windows PE Build Environment

In this step, you create a required directory structure that supports building a Windows PE image.

1. On your technician computer, click **Start**, point to **All Programs**, point to **Windows OPK** or **Windows AIK**, and then click **Windows PE 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\version\Tools, where **version** can be **Windows OPK** or **Windows AIK**.
2. At the command prompt, run the **Copype.cmd** script. The script requires two arguments: hardware architecture and destination location. For example,

```
copype.cmd <architecture> <destination>
```

   where <architecture> can be **x86**, **amd64**, or **ia64** and <destination> is a path to the local directory. For example,

```
copype.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
```

Step 2: Mount the Base Windows PE Image

In this step, you mount the base image to a local directory so that you can add or remove packages.

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

```
imagex /mountrw c:\winpe_x86\winpe.wim 1 c:\winpe_x86\mount
```

Step 3: Add Additional Packages

By using the Peimg tool, you install Windows features by using the **/install** option. Windows features are included with the base image (Winpe.wim) but are not installed. You can also import packages and add drivers and language packs. For more information, see Windows PE Customization How-To Topics.

1. Add a Windows feature to the base image by using the **peimg /install** command. For example,

```
peimg /install=<pkg> c:\winpe_x86\mount\Windows
```

where <pkg> denotes the package name. A list of available packages and their names can be obtained by using the **/list** command. You can use wildcards to specify a package name. Any packages with matching names will be installed. For example,

```
peimg /install=WinPE-HTA-Package c:\winpe_x86\mount\Windows
```

-or-

```
peimg /install=*HTA* c:\winpe_x86\mount\Windows
```

where wildcards denote any package with HTA in the package name.

Windows PE 2.0 provides the following Windows features referred to as packages.

Package Name	Description
WinPE-FONTSupport-<region>-Packages	Additional font support for ja-jp, ko-kr, zh-cn, zh-hk, and zh-tw.
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 component (available only on the Windows OPK)
WinPE-WMI-Packages	Windows Management Instrumentation (WMI) support
WinPE-XML-Package	Microsoft XML (MSXML) parser support

2. Repeat Step 1 for each package.
3. 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:\winpe_x86\mount\Windows
```

In the **INS** column, **(+)** denotes installed packages and **(-)** denotes not installed.

Step 4: (Optional) Add Additional Customizations

This step is optional but recommended. You can add applications and scripts to your Windows PE image that you might need while working in Windows PE. The following is a list of common tools to include in your Windows PE image.

- **ImageX**
  A command-line tool for capturing and applying images during deployment scenarios. For example, at a command prompt,

```
copy “c:\program files\<version>\Tools\x86\imagex.exe” c:\winpe_x86\iso\
```

- **Package Manager (Pkgmgr.exe)**
  A tool for servicing Windows image (.wim) files offline. You must copy the entire \Servicing folder and MSXML6 binaries. Offline servicing requires ImageX. For example,

```
xcopy “c:\program files\<version>\Tools\<architecture>\Servicing”
c:\winpe_x86\iso\Servicing /s
```

```
copy %windir%\system32\msxml6*.dll c:\winpe_x86\iso\Servicing
```

where <version> can be Windows OPK or Windows AIK and <architecture> can be **x86**, **amd64** or **ia64**. In both previous examples, the tools are not loaded into memory during a Windows PE RAM boot. The media must be available to access the tools.

To load the tools into memory along with Windows PE, copy the source files into the mounted \Windows directory. For example,

```
c:\winpe_x86\mount\Windows
```

Important

Adding files to the \Windows directory will increase the size of your Windows PE RAM image. Ensure that your computer has sufficient memory to boot Windows PE and to run various applications.

Step 5: Prepare the Image

In this step, you prepare the image by using the **peimg /prep** command. This operation removes any non-installed packages from the final image. This operation reduces the overall image size. For example,

```
peimg /prep c:\winpe_x86\mount\Windows
```

The **/prep** option cannot be reverted, and after the **/prep** option is run, the **/install**, **/uninstall**, **/import**, and **/list** options will not function, while the **/lang** and **/inf** options will continue to function. The Peimg tool prompts you to confirm the command. To suppress this prompt for scripting, add the **/f** option.

Step 6: Commit Changes to the Image

In this step, you commit the changes to the original image file (Winpe.wim) by using the ImageX **/unmount** option with the **/commit** option. For example,

```
imagex /unmount c:\winpe_x86\mount /commit
```

Step 7: Replace the Default Boot.wim File

In this step, you replace the default Boot.wim in the \ISO directory with your new customized image. The image must be called Boot.wim. For example,

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

Next Step

You now have a customized Windows PE RAM disk image that you can place on bootable media, like a CD-ROM or a UFD.

Step-by-Step Guide for Creating a Window PE Image

to be used as an internal resource only

To create a bootable CD-ROM

1. On your technician computer, at a command prompt, create an .iso file by using Oscdimg. For example,

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

   For IA-64 architecture, replace Etfsboot.com with Efisys.bin.

2. Burn the image (Winpe_x86.iso) to a CD-ROM.

To create a bootable UFD

1. During a running Windows Vista operation system or a Windows PE session, insert your UFD device.
2. At a command prompt, use Diskpart to format the device as FAT32 spanning the entire device, setting the partition to active. For example,

```
diskpart
select disk 1
clean
create partition primary size=<size of device>
select partition 1
active
format fs=fat32
assign
exit
```

   where the value of `disk 1` is equal to `UFD`.

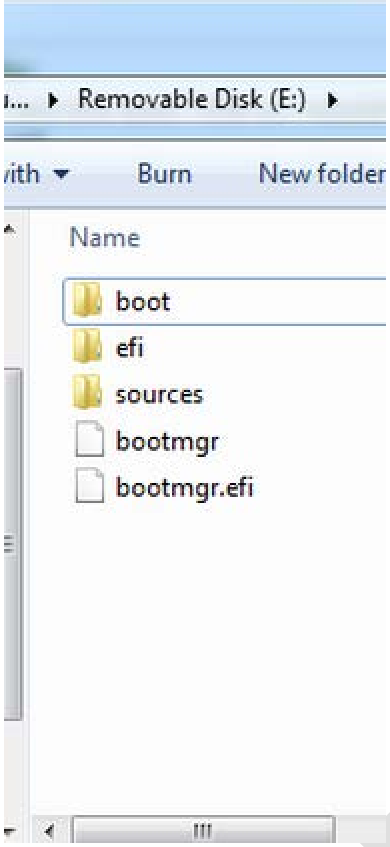
3. On your technician computer, copy all the content in the \ISO directory to your UFD device. You can manually create the directory structure or use the **xcopy** command to automatically build and copy the appropriate files from your technician computer to your UFD device. For example,

```
xcopy c:\winpe_x86\iso\*.* /s /e /f f:\
```

   where c is the letter of your technician computer hard disk and f is the letter of your UFD device.

You can further customize your Windows PE image by adding language packs, customized scripts, and drivers. For more information, see [Windows PE Customization How-To Topics]().

Your USB Drive should look like this:



Note, your .WIM goes in the sources folder, and should be named boot.wim