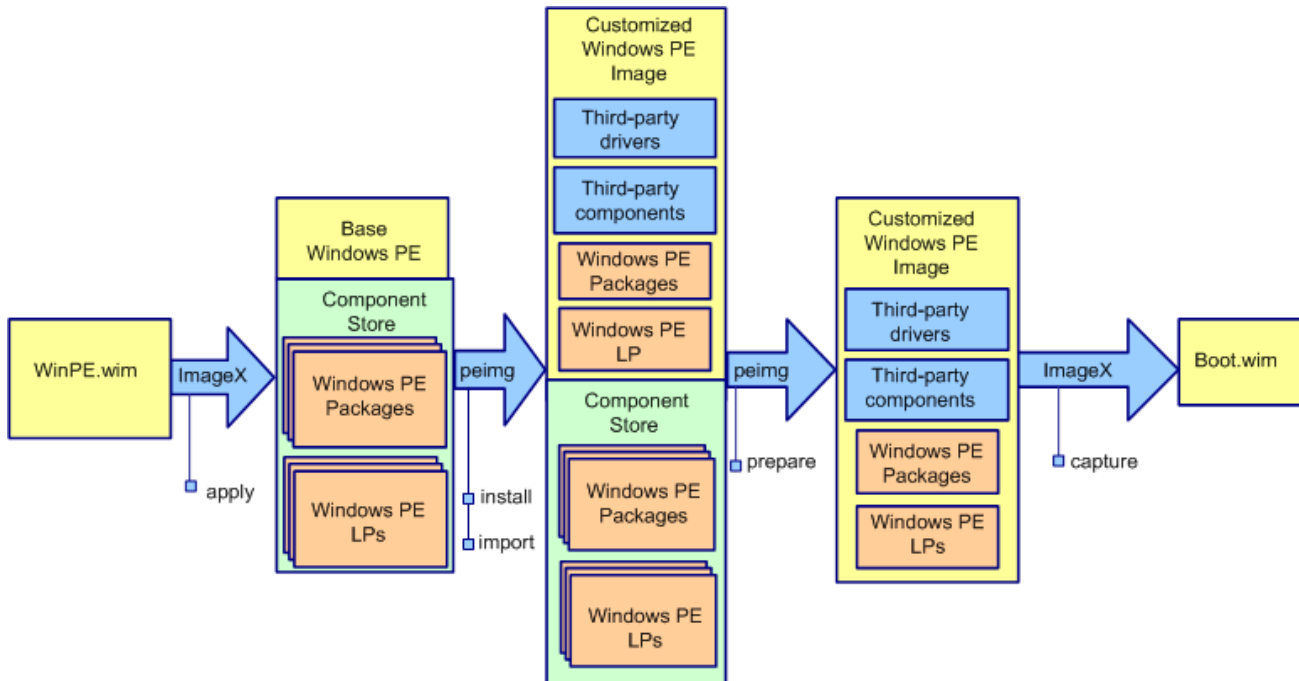


Building a Windows PE Image

The first step in creating a customized Windows PE 2.0 image is to modify the base Windows PE image (Winpe.wim) by using ImageX and PEImg tools. ImageX is required to extract the files to a local directory. PEImg enables you to add and to remove packages, and to add out-of-box drivers and language packs. ImageX then enables you to recapture changes back into a .wim file.

The following diagram illustrates how a custom Windows PE image is built.



The general process for creating a custom Windows PE image includes:
Applying the base image (Winpe.wim) by using ImageX to a local directory share. For example,

```
imagex /apply WinPE.wim 1 C:\myWinPE
```

Using the `peimg /list` command to see which packages are installed and available for installation. For example,

```
peimg /list C:\myWinPE\Windows
```

Adding drivers, packages, and language packs as appropriate by using `peimg /install` command. For example, to add the HTA package,

```
peimg /install=WinPE-HTA-Package C:\myWinPE\Windows
```

Copying any additional files or tools that you intend to include in the image. For example,

```
C:\myWinPE\Program Files\myapp\myapp.exe.
```

Preparing the image by using the `peimg /prep` command. This command will remove any packages and language packs that are not designated for the final image. For example,

Building a Windows PE Image

```
peimg /prep c:\myWinPE\Windows
```

Capturing the new bootable image by using ImageX. For example,

```
imagex /boot /compress max /capture c:\myWinPE c:\boot.wim "My Winpe Image"
```

The customized image is now ready to be copied or burned to bootable media. The most common method is to create an .iso file and then burn the image onto a CD-ROM. For more information about bootable methods, see Booting Windows PE.

Supported Packages

Windows PE 2.0 provides the following optional 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 support
WinPE-WMI-Packages	Windows Management Instrumentation (WMI) support
WinPE-XML-Package	Microsoft XML (MSMXL) Parser support

Note

The base Windows PE image (Winpe.wim) contains all the packages listed above, but they are not installed. They are only staged. You must use the PEImg tool to install the appropriate packages. When you run the peimg /prep command, packages that are not installed will be removed from the final image.

Supported Customizations

Windows PE 2.0 supports the following customizations:

- Adding and removing packages and language packs from the base image.
- Adding third-party drivers and third-party components.
- Adding Windows PE updates.
- Adding out-of-box Windows PE language packs.
- Customizing temporary storage.

Limitations

Once the peimg /prep command is applied to an image, the image is final. You can no longer add or remove packages. However, you can still add drivers and language packs.

Building a Windows PE Image

Note

It is recommended that you recapture and save a copy of the image prior to running the `peimg /prep` command. This ensures that you have a customized base image that you can later modify. You must then reapply your customized base image, run the `peimg /prep` command against the image to optimize for size, and then recapture the new image.

Customizing Windows PE Environment

You can customize your Windows PE image to include specific environment settings when Windows PE boots up. For example, you can define specific network requirements, run customized scripts, or start customized applications. You can define these settings by using a combination of `Winpeshl.ini`, `Startnet.cmd`, and `Unattend.xml`.

You can launch customized applications by using `winpeshl.ini`, define custom command-line scripts by using `Startnet.cmd`, and define various Windows PE settings and actions by using `Unattend.xml`. For more information, see [Include a Custom Script in a Windows PE Image](#).

You can define temporary storage to support customized applications. For more information, see [Add an Application to a Windows PE Image](#).

Add an Application to a Windows PE Image

The following procedure demonstrates how to add a custom application to a Windows PE image offline. For example, if you have diagnostic software or custom tools that are a core part of your validation process, you can include these tools in your custom Windows PE installation before preinstalling the operating system.

To add an application to a customized Windows PE image, create a folder for the application in the Windows PE image (such as `\Tools`) or copy the application to the `\System32` folder of the Windows PE image. If your application requires temporary storage, you can specify a custom scratch space as part of your image.

Before you can add any applications, you must first use ImageX to apply or mount the Windows PE image. Windows PE provides a base image (`Winpe.wim`) that you can customize.

For an example of how to build custom Windows PE images, see [Windows PE Walkthroughs](#). To add an application to a Windows PE image offline

Apply the base image (`Winpe.wim`) by using ImageX to a local Windows PE directory. For example,

```
imagex /apply WinPE.wim 1 c:\winpe_x86\mount\
```

or

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

Copy the necessary application files to your local Windows PE directory. For example,

```
C:\Winpe_x86\Tools to include the application on the bootable media
```

Building a Windows PE Image

or

`C:\Winpe_x86\Mount\Windows\MyApp\.`

Repeat the first two steps for each additional custom application.

To start the application automatically when Windows PE boots, you can create a startup script by using a Winpeshl.ini file. For an example, see [Include a Custom Script in a Windows PE Image](#).

If your application requires temporary storage, you can allocate additional memory using the `peimg /scratchspace` command. Valid sizes include 32, 64, 128, 256, or 512 megabytes (MB). This feature is only available offline. You cannot adjust this setting while a Windows PE session is running.

Important

Make sure you have sufficient memory to load and run your custom Windows PE image. In addition to the image size, you should have at least 256 MB of available working memory. If you have limited memory, define a page file (Pagefile.sys) to improve memory management. For more information on implementing a page file, see [Wpeutil Command-Line Options](#).

When you finish customizing the image, prepare the image for deployment by using the `peimg /prep` command.

After preparing the image, you can burn the image to media. First, recapture or commit the local Windows PE directory to a .wim file by using ImageX, create an .iso file by using one of the provided tools, and then burn the .iso file to the appropriate media.