

DISM Image Management Command-Line Options

9 out of 17 rated this helpful - [Rate this topic](#)

Published: April 1, 2014

Updated: May 30, 2014

Applies To: Windows 8, Windows 8.1, Windows Server 2012, Windows Server 2012 R2

Deployment Image Servicing and Management (DISM.exe) mounts a Windows image (.wim) file or virtual hard disk (.vhd or .vhdx) for servicing. You can also use the DISM image management command to list the image index numbers, to verify the architecture for the image that you are mounting, append an image, apply an image, capture an image and delete an image. After you update the image, you must unmount it and either commit or discard the changes that you have made.

This topic discusses DISM commands related to image management. To see other command-line options, see [Deployment Image Servicing and Management \(DISM\) Command-Line Options](#). For more information about common DISM scenarios, see [What is DISM?](#).

In addition to the command-line tool, DISM is available by using Windows PowerShell. For more information, see [Deployment Imaging Servicing Management \(DISM\) Cmdlets in Windows PowerShell](#).

[DISM Command-Line Options](#)

The following commands can be used to mount, unmount, capture, append, and delete and query .wim, .vhd and .vhdx files. These options are not case sensitive.

Option/Argument	Description
Option: /Append-Image	Adds an additional image to a .wim file.
Arguments:	/AppendImage compares new files to the resources in the existing .wim file specified by the /ImageFile argument, and stores only a single copy of each unique file so that each file is only captured once. The .wim file can have only one assigned compression type. Therefore, you can only append files with the same compression
/ImageFile: <i><path_to_image_file></i>	
/CaptureDir: <i><source_directory></i>	

/Name:*<image_name>*

[/Description:*<image_description>*]

[/ConfigFile:*<configurtion_file.ini>*]

[/Bootable]

/WIMBoot

[/CheckIntegrity]

[/Verify]

[/NoRpFix]

type.

This command-line option does not apply to virtual hard disk (VHD) files.

Important

Ensure that you have enough disk space for the **/Append-Image** option to run. If you run out of disk space while the image is being appended, you might corrupt the .wim file.

Use **/WIMBoot** to append the image with Windows image file boot (WIMBoot) configuration. This only applies to Windows 8.1 Update images that have been captured or exported as a WIMBoot file. For more information about WIMBoot, see [Windows Image File Boot \(WIMBoot\) Overview](#)

/ConfigFile specifies the location of a configuration file that lists exclusions for image capture and compress commands. For more information, see [DISM Configuration List and WimScript.ini Files](#).

/Bootable marks a volume image as being a bootable image. This argument is available only for Windows PE images. Only one volume image can be marked as bootable in a .wim file.

/CheckIntegrity detects and tracks .wim file corruption when used with capture, unmount, export, and commit operations. **/CheckIntegrity** stops the operation if DISM detects that the .wim file is corrupted when used with apply and mount operations.

/Verify checks for errors and file duplication.

/NoRpFix disables the reparse point tag fix. A reparse point is a file that contains a link to another file on the file system. If **/NoRpFix** is not specified, reparse points that resolve to paths outside of the value specified by **/ImageFile** will not be captured.

Example:

```
Dism /Append-Image /ImageFile:install.wim  
/CaptureDir:D:\ /Name:Drive-D
```

Applies an image to a specified drive.

This command-line option does not apply to virtual hard disk (VHD) files.

/CheckIntegrity detects and tracks .wim file corruption when used with capture, unmount, export, and commit operations. **/CheckIntegrity** stops the operation if DISM detects that the .wim file is corrupted when used with apply and mount operations.

/Verify checks for errors and file duplication.

/NoRpFix disables the reparse point tag fix. A reparse point is a file that contains a link to another file on the file system. If **/NoRpFix** is not specified, reparse points that resolve to paths outside the value specified by **/ImageFile** will not be captured.

/SWMFile enables you to reference split .wim files (SWMs). *pattern* is the naming pattern and location of split files. You can also specify wildcard characters. For example, "E:\image\install*.swm" will apply all of the split files in the E:\image directory named install1.swm, install2.swm, and so on.

/ConfirmTrustedFile validates the image for Trusted Desktop on a Windows 8 or Windows 8.1. This option can only be run on a computer running at least Windows Preinstallation Environment (Windows PE) 4.0.

Note

When using **/Apply-Image** with the **/ConfirmTrustedFile** option in Windows PE, always specify the **/ScratchDir** option pointed to a physical media location. This ensures that short file names will always be available. See

Option: **/Apply-Image**

Arguments:

/ImageFile:<path_to_image_file>

[/SWMFile:<pattern>]

/ApplyDir:<target_directory>

{**/Index:**< image_index> |
/Name:<image_name>}

[/CheckIntegrity]

[/Verify]

[/NoRpFix]

[/ConfirmTrustedFile]

[/WIMBoot]

[DISM Global Options for Command-Line Syntax](#) for more information about the default behavior of the **/ScratchDir** option.

Use **/WIMBoot** to apply the image with Windows image file boot (WIMBoot) configuration. This only applies to Windows 8.1 Update images that have been captured or exported as a WIMBoot file. For more information about WIMBoot, see [Windows Image File Boot \(WIMBoot\) Overview](#)

Example:

```
Dism /apply-image /imagefile:install.wim /index:1 /ApplyDir:D:\
```

```
Dism /apply-image /imagefile:install.swm /swmfile:install*.swm /index:1 /applydir:D:\
```

/Capture-CustomImage captures the incremental file changes based on the specific **install.wim** file to a new file, **custom.wim** for a WIMBoot image. You can't capture an empty directory.

The captured files are converted to pointer files. The **custom.wim** is placed in the same folder next to the **install.wim**.

For more information about the full WIMBoot deployment process, see [Deploy WIMBoot Images: If you know the size of the images upfront](#).

Important

- **/Capture-CustomImage** only captures the customization files. It can't be used to capture installation files into a new WIM.
- Keep the **install.wim** and **custom.wim** files together. Don't switch out either the **custom.wim** file or the **install.wim** file.
- You can only capture the custom image

Option: **/Capture-CustomImage**

Arguments:

/CaptureDir:*<source_directory>*

/ConfigFile:*<configuration_file.ini>*

/CheckIntegrity

/Verify

/ConfirmTrustedFile

once. Don't remove or recapture a custom.wim after capturing the incremental file changes.

/CaptureDir: specifies the directory to which the image was applied and customized.

/ConfigFile specifies the location of a configuration file that lists exclusions for image capture and compress commands. For more information, see [DISM Configuration List and WimScript.ini Files](#).

/CheckIntegrity detects and tracks .wim file corruption when used with capture, unmount, export, and commit operations. **/CheckIntegrity** stops the operation if DISM detects that the .wim file is corrupted when used with apply and mount operations.

/Verify checks for errors and file duplication.

/ConfirmTrustedFile validates the image for Trusted Desktop on a Windows 8 or Windows 8.1. This option can only be run on a computer running at least Windows Preinstallation Environment (Windows PE) 4.0.

Example:

Dism /Capture-CustomImage /CaptureDir:D:

Captures an image of a drive to a new .wim file. Captured directories include all subfolders and data. You cannot capture an empty directory. A directory must contain at least one file.

This command-line option does not apply to virtual hard disk (VHD) files.

Important

DISM will ignore extended attributes during a capture operation.

/ConfigFile specifies the location of a

Option:**/Capture-Image**

Arguments:

/ImageFile:<path_to_image_file>

/CaptureDir:<source_directory>

/Name:<image_name>

[/Description:<image_description>]

[/ConfigFile:<configuration_file.ini>]

[/**Compress**:{*max|fast|none*}]

[/**Bootable**]

[/**CheckIntegrity**]

[/**Verify**]

[/**NoRpFix**]

[/**WIMBoot**]

configuration file that lists exclusions for image capture and compress commands. For more information, see [DISM Configuration List and WimScript.ini Files](#).

/Compress specifies the type of compression used for the initial capture operation. The **maximum** option provides the best compression, but takes more time to capture the image. The **fast** option provides faster image compression, but the resulting files are larger than those compressed by using the **maximum** option. This is also the default compression type that is used if you do not specify the argument. The **none** option does not compress the captured image at all.

/Bootable marks a volume image as being a bootable image. This argument is available only for Windows PE images. Only one volume image can be marked as bootable in a .wim file.

/CheckIntegrity detects and tracks .wim file corruption when used with capture, unmount, export, and commit operations. **/CheckIntegrity** stops the operation if DISM detects that the .wim file is corrupted when used with apply and mount operations.

/Verify checks for errors and file duplication.

/NoRpFix disables the reparse point tag fix. A reparse point is a file that contains a link to another file on the file system. If **/NoRpFix** is not specified, reparse points that resolve to paths outside the value specified by **/ImageFile** will not be captured.

Use **/WIMBoot** to capture the image with Windows image file boot (WIMBoot) configuration. This only applies to Windows 8.1 Update images that you want to capture as a WIMBoot file. For more information about WIMBoot, see [Windows Image File Boot \(WIMBoot\) Overview](#)

Option: **/Cleanup-Mountpoints**

Option: **/Commit-Image**

Arguments:

/MountDir:*<path_to_mount_directory>*

[/CheckIntegrity]

[/Append]

Option: **/Delete-Image**

Arguments:

/ImageFile:*<path_to_image_file>*

{/Index:*<image_index>* **|**
/Name:*<image_name>***}**

Example:

**Dism /Capture-Image /ImageFile:install.wim
/CaptureDir:D:\ /Name:Drive-D**

Deletes all of the resources associated with a mounted image that has been corrupted. This command will not unmount images that are already mounted, nor will it delete images that can be recovered using the **/Remount-Image** command.

Example:

Dism /Cleanup-Mountpoints

Applies the changes that you have made to the mounted image. The image remains mounted until the **/Unmount-Image** option is used.

/CheckIntegrity detects and tracks .wim file corruption when used with capture, unmount, export, and commit operations. **/CheckIntegrity** stops the operation if DISM detects that the .wim file is corrupted when used with apply and mount operations.

/Append adds the modified image to the existing .wim file instead of overwriting the original image.

The **/CheckIntegrity** and **/Append** arguments do not apply to virtual hard disk (VHD) files.

Example:

**Dism /Commit-Image
/MountDir:C:\test\offline**

Deletes the specified volume image from a .wim file that has multiple volume images. This option deletes only the metadata entries and XML entries. It does not delete the stream data and does not optimize the .wim file.

This command-line option does not apply to virtual hard disk (VHD) files.

[/CheckIntegrity]

/CheckIntegrity detects and tracks .wim file corruption when used with capture, unmount, export, and commit operations. **/CheckIntegrity** stops the operation if DISM detects that the .wim file is corrupted when used with apply and mount operations.

Example:

**Dism /Delete-Image /ImageFile:install.wim
/Index:1**

Exports a copy of the specified image to another file. The source and destination files must use the same compression type. You can also optimize an image by exporting to a new image file. When you modify an image, DISM stores additional resource files that increase the overall size of the image. Exporting the image will remove unnecessary resource files.

Option: **/Export-Image**

Arguments:

**{/SourceImageFile:<path_to_image_file> |
/SWMFile:<pattern>}**

**{/SourceIndex:<image_index> |
/SourceName:<image_name>}**

/DestinationImageFile:<path_to_image_file>

[/DestinationName:<Name>]

[/Compress:{fast|max|none|recovery}]

[/Bootable]

[/WIMBoot]

[/CheckIntegrity]

This command-line option does not apply to virtual hard disk (VHD) files.

/SWMFile enables you to reference split .wim files. *pattern* is the naming pattern and location of split files. You can also specify wildcard characters. For example, "E:\image\install*.swm" will export the split files in the E:\image directory named install1.swm, install2.swm, and so on.

/Compress specifies the type of compression used for the initial capture operation. The **/Compress** argument does not apply when you export an image to an existing .wim file, you can only use this argument when you export an image to a new .wim file. The **maximum** option provides the best compression, but takes more time to capture the image. The **fast** option provides faster image compression, but the resulting files are larger than those compressed by using the **maximum** option. This is also the default compression type that is used if you do not specify the argument. Use the **recovery** option to export push-button reset images. The resulting files are much smaller in size, which in

turn, greatly reduce the amount of disk space needed for saving the push-button reset image on a recovery drive. The destination file must be specified with an **.esd** extension. The **none** option does not compress the captured image at all.

/Bootable marks a volume image as being a bootable image. This argument is available only for Windows PE images. Only one volume image can be marked as bootable in a .wim file.

Use **/WIMBoot** to apply the image with Windows image file boot (WIMBoot) configuration. This only applies to Windows 8.1 Update images that you want to export as a WIMBoot file. For more information about WIMBoot, see [Windows Image File Boot \(WIMBoot\) Overview](#)

/CheckIntegrity detects and tracks .wim file corruption when used with capture, unmount, export, and commit operations. **/CheckIntegrity** stops the operation if DISM detects that the .wim file is corrupted when used with apply and mount operations.

Example:

**Dism /Export-Image
/SourceImageFile:install.wim /SourceIndex:1
/DestinationImageFile:install2.wim**

Lists the images that are currently mounted and information about the mounted image such as whether the image is valid, read/write permissions, mount location, mounted file path, and mounted image index.

Example:

Dism /Get-MountedImageInfo

Displays information about the images that are contained in the .wim, vhd or .vhdx file. When used with the **/Index** or **/Name** argument, information about the specified image is displayed, which includes if an image is a

Option: **/Get-MountedImageInfo**

Option: **/Get-ImageInfo**

Arguments:

/ImageFile:*<path_to_image.wim>*

[**/Index:***<Image_index>* |
/Name:*<Image_name>*]

WIMBoot image, if the image is Windows 8.1 Update, see [Take Inventory of an Image or Component Using DISM](#). The **/Name** argument does not apply to VHD files. You must specify **/Index:1** for VHD files.

Example:

Dism /Get-ImageInfo
/ImageFile:C:\test\offline\install.wim

Dism /Get-ImageInfo
/ImageFile:C:\test\images\myimage.vhd
/Index:1

Use **/Get-WIMBootEntry** to display WIMBoot configuration entries for the specified disk volume.

For more information about how to display WIMBoot configuration entries, see [Take Inventory of an Image or Component Using DISM](#).

Option: **/Get-WIMBootEntry**

Arguments:

/Path:*<volume_path>*

This only applies to Windows 8.1 Update. For more information about WIMBoot, see [Windows Image File Boot \(WIMBoot\) Overview](#)

Example:

Dism /Get-WIMBootEntry /Path:C:
Displays a list of the files and folders in a specified image.

Option: **/List-Image**

Arguments:

/ImageFile:*<path_to_image_file>*

{**/Index:***<image_index>* |
/Name:*<image_name>*}

This command-line option does not apply to virtual hard disk (VHD) files.

Example:

Dism /List-Image /ImageFile:install.wim
/Index:1

Mounts an image from a .wim, .vhd or .vhdx file to the specified directory so that it is available for servicing.

Option: **/Mount-Image**

Arguments:

/ReadOnly sets the mounted image with read-

/ImageFile:<path_to_image>

{**/Index:**<image_index> |
/Name:<image_name> }

/MountDir:<path_to_mount_directory>

[**/ReadOnly**]

[**/Optimize**]

[**/CheckIntegrity**]

only permissions. Optional.

Note

An index or name value is required for most operations that specify a .wim file.

You can use **/Optimize** to reduce initial mount time.

Important

When using the **/optimize** argument, processes that are ordinarily performed during a mount will instead be completed the first time that you access a directory. As a result, there may be an increase in the time that is required to access a directory for the first time after mounting an image using the **/optimize** argument.

/CheckIntegrity detects and tracks .wim file corruption when used with capture, unmount, export, and commit operations. **/CheckIntegrity** stops the operation if DISM detects that the .wim file is corrupted when used with apply and mount operations.

Example:

Dism /Mount-Image

/ImageFile:C:\test\images\myimage.wim
/index:1 /MountDir:C:\test\offline

Dism /Mount-Image

/ImageFile:C:\test\images\myimage.vhd
/index:1 /MountDir:C:\test\offline /ReadOnly

Performs specified configurations to an offline image.

Use **/WIMBoot** to configure an offline image for installing on a Windows image file boot (WIMBoot) system.

/Optimize-Image /WIMBoot

/Optimize-Image /WIMBoot only applies to Windows 8.1 Update images that have been captured or exported as a WIMBoot file. For more information about WIMBoot, see [Windows](#)

[Image File Boot \(WIMBoot\) Overview.](#)

Important

Only use **/Optimize-Image** with images that will be used for WIMBoot supported systems. If **/Optimize-Image** is used with a non-WIMBoot supported system image, Windows may not work as expected, after installation on a non-WIMBoot supported device.

Examples:

DISM.exe /Image:C:\test\offline /Optimize-Image /WIMBoot

Remounts a mounted image that has become inaccessible and makes it available for servicing.

Example:

Dism /Remount-Image /MountDir:C:\test\offline

Splits an existing .wim file into multiple read-only split .wim files.

This option creates the .swm files in the specified directory, naming each file the same as the specified *path_to_swm*, but with an appended number. For example, if you set *path_to_swm* as `c:\Data.swm`, this option creates a Data.swm file, a Data2.swm file, a Data3.swm file, and so on, defining each portion of the split .wim file and saving it to the C:\ directory.

This command-line option does not apply to virtual hard disk (VHD) files.

/FileSize specifies the maximum size in megabytes (MB) for each created file.

Note

If a single file is larger than the value specified in the **/FileSize** option, one of the split .swm files that results will be larger than the value specified in the **/FileSize** option, in order to accommodate the large file.

Option: **/Remount-Image**

Arguments:

/MountDir:*<path_to_mount_directory>*

Option: **/Split-Image**

Arguments:

/ImageFile:*<path_to_image_file>*

/SWMFile:*<path_to_swm>*

/FileSize:*<MB-Size>*

[/CheckIntegrity]

/CheckIntegrity detects and tracks .wim file corruption when used with capture, unmount, export, and commit operations. **/CheckIntegrity** stops the operation if DISM detects that the .wim file is corrupted when used with apply and mount operations.

Example:

Dism /Split-Image /ImageFile:install.wim /SWMFile:split.swm /FileSize:650

Unmounts the .wim, .vhd or .vhdx file and either commits or discards the changes that were made when the image was mounted.

You must use either the **/commit** or **/discard** argument when you use the **/Unmount-Image** option.

/CheckIntegrity detects and tracks .wim file corruption when used with capture, unmount, export, and commit operations. **/CheckIntegrity** stops the operation if DISM detects that the .wim file is corrupted when used with apply and mount operations.

/Append adds the modified image to the existing .wim file instead of overwriting the original image.

The **/CheckIntegrity** and **/Append** arguments do not apply to virtual hard disk (VHD) files.

Example:

Dism /Unmount-Image /MountDir:C:\test\offline /commit

Dism /Unmount-Image /MountDir:C:\test\offline /discard

Use **/Update-WIMBootEntry** Updates the WIMBoot configuration entry, associated with the specified data source ID, with the renamed image file or moved image file path.

Option: **/Unmount-Image**

Arguments:

/MountDir:<path_to_mount_directory>

{**/Commit** | **/Discard**}

[**/CheckIntegrity**]

[**/Append**]

Option: **/Update-WIMBootEntry**

Arguments:

/Path:<Volume_path>

/DataSourceID:<Data_source_id>

/ImageFile:<Renamed_image_path>

Note

/Update-WIMBootEntry requires a restart in order for any updates to take effect.

Use **/Path** to specify the disk volume of the WIMBoot configuration.

Use **/DataSourceID** to specify the data source ID as displayed by **/Get-WIMBootEntry**.

This only applies to Windows 8.1 Update. For more information about WIMBoot, see [Windows Image File Boot \(WIMBoot\) Overview](#)

Example:

```
DISM.exe /Update-WIMBootEntry /Path:C:\  
/DataSourceID:0 /ImageFile:R:\Install.wim
```