

Windows PE

Windows Preinstallation Environment (Microsoft Corporation)

What is Windows PE?

Windows Preinstallation Environment (Windows PE) 2.0 is a minimal Win32 operating system with limited services, built on the Windows Vista kernel. It is used to prepare a computer for Windows installation, to copy disk images from a network file server, and to initiate Windows Setup.

Windows PE is not designed to be the primary operating system on a computer, but is instead used as a standalone preinstallation environment and as an integral component of other setup and recovery technologies, such as Setup for Windows Vista, Windows Deployment Services (Windows DS), the Systems Management Server (SMS) Operating System (OS) Deployment Feature Pack, and the Windows Recovery Environment (Windows RE).

This topic includes:

- Benefits of Windows PE
- Common Windows PE Scenarios
- Windows PE Limitations
- Windows PE Dependencies
- Supported Drivers
- Technologies Related to Windows PE

Benefits of Windows PE

Windows PE was created to help original equipment manufacturers (OEMs) and corporations boot a computer with no functioning operating system. In the past, OEMs and corporations often used an MS-DOS-based boot floppy disk to start a computer, and then either to:

- Connect to a network share where a Windows installation source or disk image was located.
- Troubleshoot and recover a copy of Windows that did not start.

However, an MS-DOS-based boot floppy disk has a number of limitations that make it awkward to use for preinstalling Windows or recovering existing installations or data:

- No support for the NTFS file system.
- No native networking support.
- No support for 32-bit (or 64-bit) Windows device drivers, making it necessary to locate 16-bit drivers.
- Rudimentary support for custom applications and scripts.

The limitations of MS-DOS-based boot disks led Microsoft to develop Windows PE, which is Microsoft's primary tool for booting computers with no functional operating system. Once you boot a computer into Windows PE, you can prepare a computer for Windows installation and then initiate Windows Setup from a network or a local source. Otherwise, you can service an existing copy of Windows or recover data.

Because Windows PE is based on the kernel for Windows Vista, it overcomes the limitations of MS-DOS-based boot disks by providing the following capabilities:

- Native support for NTFS 5.x file systems, including dynamic volume creation and management.

Windows PE

Windows Preinstallation Environment (Microsoft Corporation)

- Native support for TCP/IP networking and file sharing (client only).
- Native support for 32-bit (or 64-bit) Windows device drivers.
- Native support for a subset of the Win32 Application Programming Interface (API); optional support for Windows Management Instrumentation (WMI) and Windows Script Host (Windows SH).
- Can be started from different kinds of media, including CDs, DVDs, USB flash devices (UFD), and Windows Deployment Services (Windows DS).

NOTE: This section discusses the standalone version of Windows PE. Customized versions of Windows PE that are used in other products and Windows technologies (such as Windows DS) may provide different functionality. For more information about using Windows PE in these environments, consult the relevant product documentation.

Common Windows PE Scenarios

Windows PE is a modified version of the Windows operating system that is designed to support installing Windows and troubleshooting and recovering an installation that can no longer boot.

- Installing Windows Vista. Windows PE runs every time you install Windows Vista. The graphical tools that collect configuration information during the setup phase are running within Windows PE. In addition, information technology (IT) departments can customize and extend Windows PE to meet their unique deployment needs. Windows PE also provides support for servicing Windows images.

IMPORTANT: Windows PE supports the deployment of previous versions of Windows. Refer to your licensing agreement for restrictions or contact your Microsoft representative.

- Troubleshooting. Windows PE is useful for both automatic and manual troubleshooting. For example, if Windows Vista fails to start because of a corrupted system file, Windows PE can automatically start and launch the Windows Recovery Environment (Windows RE). You can also manually start Windows PE to use built-in or customized troubleshooting and diagnostic tools.
- Recovery. OEMs and independent software vendors (ISVs) can use Windows PE to build customized, automated solutions for recovering and rebuilding computers running Windows Vista. For example, users can start their computers from Windows PE recovery CDs or recovery partitions to automatically reformat their hard disks and to reinstall Windows Vista with the original drivers, settings, and applications.

Windows PE Limitations

Windows PE is a subset of Windows Vista, and has the following limitations:

- To reduce its size, Windows PE includes only a subset of the available Win32 application programming interfaces (APIs). I/O (disk and network) and core Win32 APIs are included.
- To prevent its use as a pirated operating system, Windows PE automatically stops running the shell and reboots after 72 hours of continuous use. This time period is not configurable.

Windows PE

Windows Preinstallation Environment (Microsoft Corporation)

- Windows PE cannot act as a file server or terminal server. (Remote Desktop is unsupported.)
- Distributed File System (DFS) name resolution is supported for standalone roots only. Domain roots are not supported.
- The tested methods of gaining network connectivity to file servers are TCP/IP and NetBIOS over TCP/IP. Other methods, such as the IPX/SPX network protocol, are not supported.
- All changes that you make to the Windows PE registry while running Windows PE are lost the next time you restart the computer. To make permanent registry changes, you must edit the registry offline before starting Windows PE.
- Drive letters are assigned in consecutive order as you create partitions in Windows PE. However, the drive letters are reset to the default order when you restart Windows PE.
- Windows PE does not support the Microsoft .NET Framework or the Common Language Runtime (CLR).
- Windows PE does not include the "Windows on Windows 32" (WOW32), "Windows on Windows 64" (WOW64), Virtual DOS Machine (VDM), OS/2 or POSIX subsystems.
- To install a 64-bit version of Windows you must use a 64-bit version of Windows PE. Likewise, to install a 32-bit version of Windows, you must use a 32-bit version of Windows PE.
- Windows PE can be used to configure and to partition a computer's disks before starting Windows Setup. If any hard disks are converted to dynamic disks with Diskpart.exe before you start Windows Setup, then those hard disks are recognized as "foreign" when the operating system is installed, and any volumes on those hard disks will not be accessible.
- Windows PE does not support applications packaged with Windows Installer (.msi).
- WIM File System Filter (WIM FS Filter) driver is not supported in Windows PE.

Windows PE Dependencies

Windows PE depends on a number of Windows technologies to function properly.

- If you use Windows PE on a network, DHCP and DNS servers are useful, but not required.
- If you boot Windows PE from the network, you must use a Windows DS server.
- If you install Windows, you must run Windows Setup (Setup.exe in Windows Vista or Winnt32.exe for Windows Server 2003).
- If you automate Windows Vista installation, you must use an Unattend.xml answer file.

Supported Drivers

Windows PE 2.0 supports the following drivers:

1394.inf
acpi.inf

adp94xx.inf
adpahci.inf

adpu160m.inf
adpu320.inf

Windows PE

Windows Preinstallation Environment
(Microsoft Corporation)

Unattended Setup is particularly useful if you need to:

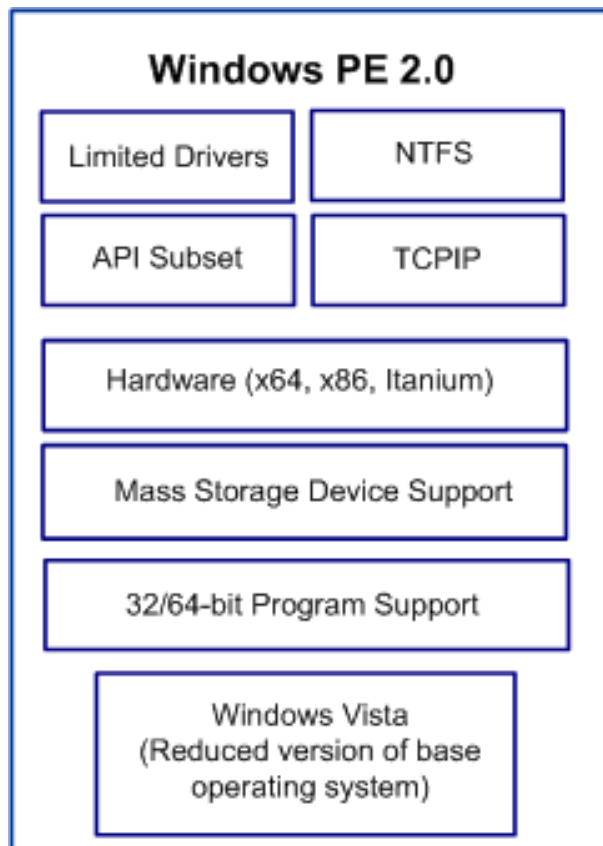
- Perform automated installations on computers that have heterogeneous hardware configurations.
- Perform automated installations on specific types of servers, such as domain controllers, remote access servers, and servers that run Certificate Services or the Cluster service.
- Configure a wide range of operating system settings during an automated installation without using batch files and scripts.

In addition to these deployment solutions, unattended Setup is a useful method of creating master installations for image-based and RIS installations.

A common method to initiate an unattended Setup is to boot the computer by using Windows PE, and then launch Setup with an answer file from a distribution folder on the network.

How Windows PE Works

Windows PE version 2.0 is a minimal Win32 subsystem with limited services, based on the Windows Vista kernel running in protected mode.



Windows PE Optimal Environment

An optimal environment for running Windows PE has the following characteristics:

Windows PE

Windows Preinstallation Environment (Microsoft Corporation)

- Sufficient RAM for RAM disk boot.
- If you choose to use a USB Flash Drive (UFD) device, it must meet all the requirements for booting Windows PE.
- The appropriate network adapter and mass storage drivers available, either on the Windows PE CD or available to place on a customized Windows PE image.
- Physical access to the computer, or, if using Emergency Management Services (EMS) to use Windows PE in the configuration of a remotely administered server, the computer properly supports EMS.
- If you choose to use Windows DS, the server must run Windows Server 2003 RIS server, and the destination computers must have a network adapter that enables PXE or one that is supported by the Windows DS start disk.

Order of Operations in Windows PE

The boot process of Windows PE is as follows.

1. The boot sector on the particular media is loaded. Control is passed to Bootmgr. Bootmgr extracts basic boot information from the Boot Configuration Data (BCD) and passes control to winload.exe that is contained in Boot.wim. Winload.exe then loads the appropriate Hardware Abstraction Layer (HAL), and loads the System registry hive and necessary boot drivers. After it finishes loading, it prepares the environment to execute the kernel, Ntoskrnl.exe.

NOTE: If you start Windows PE from read-only media such as a CD, Windows PE stores the registry hives in memory so that applications can write to the registry. Any changes made to the registry by the applications do not persist across different Windows PE sessions.

2. Ntoskrnl.exe is executed and finishes the environment setup. Control is passed to the Session Manager (SMSS).
3. SMSS loads the rest of the registry, configures the environment to run the Win32 subsystem (Win32k.sys) and its various processes. SMSS loads the Winlogon process to create the user session, and then starts the services and the rest of the non-essential device drivers and the security subsystem (LSASS).
4. Winlogon.exe runs setup based on the registry value HKLM\SYSTEM\Setup\CmdLine. Wimpeshl.exe will launch %SYSTEMDRIVE%\sources\setup.exe if it exists, otherwise it looks for an application specified in %SYSTEMROOT%\system32\wimpeshl.ini. If no application is specified, Wimpeshl.exe will execute cmd /k %SYSTEMROOT%\system32\startnet.cmd. By default, Windows PE contains a Startnet.cmd file which will launch Wpeinit.exe. Wpeinit.exe loads network resources and coordinates with networking components like DHCP.
5. When Wpeinit.exe completes, the Command Prompt window is displayed. The boot process of Windows PE is complete.

Windows PE Tools and Settings

This topic describes the tools and the settings that are related to the Windows Preinstallation Environment Windows Preinstallation Environment 2.0 (Windows PE 2.0).

Windows PE

Windows Preinstallation Environment (Microsoft Corporation)

Windows PE Command-Line Tools

Windows PE includes the following command-line tools:

- BCDEdit
- Bootsect
- Oscdimg
- DiskPart
- Drvload
- PEImg
- Wpeinit

BCDEdit

Boot Configuration Data (BCD) files provide a store that is used to describe boot applications and boot application settings. The objects and elements in the store effectively replace Boot.ini.

BCDEdit is a command-line tool for managing BCD stores. It can be used for a variety of purposes, including creating new stores, modifying existing stores, adding boot menu options, and so on.

BCDEdit serves essentially the same purpose as Bootcfg.exe on earlier versions of Windows, but with two major improvements:

- BCDEdit exposes a wider range of boot options than Bootcfg.exe.
- BCDEdit has improved scripting support.

NOTE: Administrative privileges are required to use BCDEdit to modify BCD.

BCDEdit is the primary tool for editing the boot configuration of Windows Vista and later versions of Windows. It is included with the Windows Vista distribution in the %WINDIR%\System32 folder.

BCDEdit is limited to the standard data types and is designed primarily to perform single common changes to BCD. For more complex operations or nonstandard data types, consider using the BCD Windows Management Instrumentation (WMI) application programming interface (API) to create more powerful and flexible custom tools.

BCDEdit Command-Line Options

The following command-line options are available for BCDEdit.exe.

BCDEdit/Command [Argument1] [Argument2] ...

General BCDEdit Command-Line Option

Option	Description
/?	Displays a list of BCDEdit commands. Running this command without an argument displays a summary of the available commands. To display detailed help for a particular command, run bcdedit /? command, where command is the name of the command you are searching for more information about. For example, bcdedit /? createstore displays detailed help for the Createstore

Windows PE
 Windows Preinstallation Environment
 (Microsoft Corporation)

	command.
--	----------

BCDEdit Command-Line Options that Operate on a Store

Option	Description
/createstore	Creates a new empty boot configuration data store. The created store is not a system store.
/export	Exports the contents of the system store into a file. This file can be used later to restore the state of the system store. This command is valid only for the system store.
/import	Restores the state of the system store by using a backup data file previously generated by using the /export option. This command deletes any existing entries in the system store before the import takes place. This command is valid only for the system store.
/store	This option can be used with most BCDEdit commands to specify the store to be used. If this option is not specified, then BCDEdit operates on the system store. Running the bcdedit /store command by itself is equivalent to running the bcdedit /enum active command.

BCDEdit Command-Line Options that Operate on Entries in a Store

Option	Description
/copy	Makes a copy of a specified boot entry in the same system store.
/create	Creates a new entry in the boot configuration data store. If a well-known identifier is specified, then the /application, /inherit, and /device options cannot be specified. If an identifier is not specified or not well known, an /application, /inherit, or /device option must be specified.
/delete	Deletes an element from a specified entry.

BCDEdit Command-Line Options that Operate on Entry Options

Option	Description
/deletevalue	Deletes a specified element from a boot entry.
/set	Sets an entry option value.

BCDEdit Command-Line Options that Control Output

Option	Description
/enum	Lists entries in a store. The /enum option is the default value for BCEdit, so running the bcdedit command without options is equivalent to running the bcdedit /enum active command.
/v	Verbose mode. Usually, any well-known entry identifiers are represented by their friendly

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

shorthand form. Specifying /v as a command-line option displays all identifiers in full. Running the bcdedit /v command by itself is equivalent to running the bcdedit /enum active /v command.

BCDEdit Command-Line Options that Control the Boot Manager

Option	Description
/bootsequence	Specifies a one-time display order to be used for the next boot. This command is similar to the /displayorder option, except that it is used only the next time the computer starts. Afterwards, the computer reverts to the original display order.
/default	Specifies the default entry that the boot manager selects when the timeout expires.
/displayorder	Specifies the display order that the boot manager uses when displaying boot options to a user.
/timeout	Specifies the time to wait, in seconds, before the boot manager selects the default entry.
/toolsdisplayorder	Specifies the display order for the boot manager to use when displaying the Tools menu.

BCDEdit Command-Line Options that Control Emergency Management Services

Option	Description
/bootems	Enables or disables Emergency Management Services (EMS) for the specified entry.
/ems	Enables or disables EMS for the specified operating system boot entry.
/emssettings	Sets the global EMS settings for the computer. /emssettings does not enable or disable EMS for any particular boot entry.

BCDEdit Command-Line Options that Control Debugging

Option	Description
/bootdebug	Enables or disables the boot debugger for a specified boot entry. Although this command works for any boot entry, it is effective only for boot applications.
/dbgsettings	Specifies or displays the global debugger settings for the system. This command does not enable or disable the kernel debugger; use the /debug option for that purpose. To set an individual global debugger setting, use the bcdedit /setdbgsettings type value command.
/debug	Enables or disables the kernel debugger for a specified boot entry.

To troubleshoot a new installation, enable debug mode by modifying the boot configuration file (BCD). For example, use the following syntax to enable kernel or boot debug.

```
bcdedit /set <id> debug on
```

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

-or-

```
bcdedit /set <id> bootdebug on
```

where <id> is the GUID of the Loader object that is used to load the operating system.

"Default" can be used if the operating system is the default option of the Boot Manager menu.

Bootsect

Bootsect.exe updates the master boot code for hard disk partitions to switch between BOOTMGR and NTLDR. You can use this tool to restore the boot sector on your computer. This tool replaces FixFAT and FixNTFS.

Bootsect Commands

Bootsect uses the following conventions:

```
bootsect.exe {/help | /nt52 | /nt60} {SYS | ALL | <DriveLetter:>} [/force]
```

For example, to apply the master boot code that is compatible with NTLDR to the volume labeled E, use the following command: `bootsect.exe /nt52 E:`

Command	Description
/help	Displays these usage instructions.
/nt52	Applies the master boot code that is compatible with NTLDR to SYS, ALL, or <DriveLetter>. The operating system installed on SYS, ALL, or <DriveLetter> must be a previous version of Windows Vista.
/nt60	Applies the master boot code that is compatible with BOOTMGR to SYS, ALL, or <DriveLetter>. The operating system installed on SYS, ALL, or <DriveLetter> must be Windows Vista.
SYS	Updates the master boot code on the system partition used to boot Windows.
ALL	Updates the master boot code on all partitions. ALL does not necessarily update the boot code for each volume. Instead, this option updates the boot code on volumes that can be used as Windows boot volumes, which excludes any dynamic volumes that are not connected with an underlying disk partition. This restriction is present because boot code must be located at the beginning of a disk partition.
<DriveLetter>	Updates the master boot code on the volume associated with this drive letter. Boot code will not be updated if either 1) <DriveLetter> is not associated with a volume or 2) <DriveLetter> is associated with a volume not connected to an underlying disk partition.
/force	Forcibly dismounts the volume(s) during the boot code update. You must use this option with caution. If Bootsect.exe cannot gain exclusive volume access, then the file system may overwrite the boot code before the next reboot. Bootsect.exe always attempts to lock and dismount the volume before each update. When /force is specified, a forced dismount is attempted if the

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

initial lock attempt fails. A lock can fail, for example, if files on the destination volume are currently opened by other programs.
 When successful, a forced dismount enables exclusive volume access and a reliable boot code update even though the initial lock failed. At the same time, a forced dismount invalidates all open handles to files on the destination volume. This can result in unexpected behavior from the programs that opened these files. Therefore, use this option with caution.

Oscdimg

Oscdimg is a command-line tool for creating an image file (.iso) of a customized 32-bit or 64-bit version of Windows PE. You can then burn that .iso file to a CD-ROM or DVD-ROM. Oscdimg supports ISO 9660, Joliet, and Universal Disk Format (UDF) file systems.

NOTE: Merely copying the .iso file to the CD-ROM will not work. To create a bootable CD-ROM, you must use CD-recording software that unpacks the .iso file and burns it to a CD-ROM

Oscdimg Command-Line Options

The following command-line options are available for Oscimg.

`oscdimg [options] SourceLocationTargetFile`

Option	Description
-a	Displays the allocation summary for files and directories.
-blocation	Specifies the location of the EI Torito boot sector file. Do not use any spaces, for example, -bC:\Directory\Etfsboot.com
-c	Specifies to use ANSI file names instead of OEM file names.
-d	Does not force lowercase file names to uppercase.
-e	This option means not to use floppy disk emulation in the EI Torito catalog. This option can only be used for single boot entry images and can not be combined with any multi-boot entry switches.
-g	Uses the Universal Coordinated Time (UCT) for all files rather than the local time.
-h	Includes hidden files and directories.
-j1	Encodes Joliet Unicode file names and generates DOS-compatible 8.3 file names in the ISO 9660 namespace. These file names can be read by either Joliet systems or conventional ISO 9660 systems, but Oscdimg may change some of the file names in the ISO 9660 name space to comply with DOS 8.3 and/or ISO 9660 naming restrictions. When using the -j1, -j2, or -js options, the -d, -n, and -nt options do not apply and cannot be used.
-j2	Encodes Joliet Unicode file names without standard ISO 9660 names. (Requires a Joliet operating system to read files from the CD-ROM.) When using the -j1, -j2, or -js options, the -d, -n, and -nt options do not apply and cannot be used.

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

-js	Overrides the default text file used with the -j2 option, for example, -jsc:\Readme.txt When using the -j1, -j2, or -js options, the -d, -n, and -nt options do not apply and cannot be used.
-k	Creates an image even if it fails to open some of the source files.
-llabelname	Specifies the volume label. Do not use spaces between the l and labelname. For example, -lMYLABEL
-m	Ignores the maximum size limit of an image.
-maxsize:limit	Overrides the default maximum size of an image. The default value is a 74-minute CD, unless UDF is used, in which case, the default is that there is no maximum size. The limit value is specified in megabytes (MB). For example, -maxsize:4096 limits the image to 4096 MB. The -m option cannot be used with this option.
-n	Enables long file names.
-nt	Enables long file names that are compatible with Windows NT 3.51.
-o	Optimizes storage by encoding duplicate files only once using a MD5 hashing algorithm to compare files.
-oc	Optimizes storage by encoding duplicate files only once using a binary comparison of each file. This option is slower than -o.
-oi	Optimizes storage by encoding duplicate files only once. When comparing files, ignores Diamond compression timestamps.
-os	Optimizes storage by encoding duplicate files only once. Shows duplicate files when creating the image.
-ois	Optimizes storage by encoding duplicate files only once. When comparing files, ignores Diamond compression timestamps. Shows duplicate files when creating the image.
-p	Specifies the value to use for the Platform ID in the EI Torito catalog. The default is 0x00 to represent the x86 platform. This option can only be used for single boot entry images and can not be combined with any multi-boot entry switches.
-q	Scans the source files only; it does not create an image.
-tmm/dd/yyyy,hh:mm:ss	Specifies the timestamp for all files and directories. Do not use any spaces. Use the United States of America date format and a 24-hour clock. You can use any delimiter between the items. For example, -t12/31/2000,15:01:00
-u1	Produces an image that has both the UDF file system and the ISO 9660 file system. The ISO 9660 file system will be written with DOS-compatible 8.3 file names. The UDF file system will be written with Unicode file names. This option cannot be combined with the -n, -nt, or -d options.

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

-u2	Produces an image that has only the UDF file system on it. Any system not capable of reading UDF will only see a default text file alerting the user that this image is only available on computers that support UDF. This option cannot be combined with the -n, -nt, or -d options.
-ur	Overrides the default text file used with the -u2 option. Example: -urc:\Readme.txt This option cannot be combined with the -n, -nt, or -d options.
-us	Creates sparse file when available. This can only be used with the -u2 option. This option cannot be combined with the -n, -nt, or -d options.
-ue	Creates embedded files. This can only be used with the -u2 option. This option cannot be combined with the -n, -nt, or -d options.
-uf	Embeds UDF file identifier entries. This can only be used with the -u2 option. This option cannot be combined with the -n, -nt, or -d options.
-uv	Specifies UDF Video Zone compatibility during DVD Video/Audio disk creation. This means UDF 1.02 and ISO 9660 are written to the disk. Also, all files in the VIDEO_TS, AUDIO_TS, and JACKET_P directories are written first. These directories take precedence over all other ordering rules used for this image. This option cannot be combined with the -n, -nt, -d, -j1, -j2, -js, -u1, -u2, -ur, -us, -ue, -uf, or -yl options.
-ut	Truncates the ISO 9660 portion of the image during DVD video/audio disk creation. When this option is used, only the VIDEO_TS, AUDIO_TS, and JACKET_P directories are visible from the ISO 9660 file system. This option cannot be combined with the -n, -nt, -d, -j1, -j2, -js, -u1, -u2, -ur, -us, -ue, -uf, or -yl options.
-w1	Reports all file names or depths that are not ISO-compliant or Joliet-compliant.
-w2	Reports all file names that are not DOS-compliant.
-w3	Reports all zero-length files.
-w4	Reports each file name that is copied to the image.
-x	Computes and encodes the AutoCRC value in the image.
-yd	Suppresses warnings for non-identical files with the same initial 64,000 bytes.
-yl	Specifies long allocation descriptors instead of short allocation descriptors. This option cannot be combined with the -n, -nt, or -d options.
-y5	Specifies file layout on disk. This option writes all files in an i386 directory first and in reverse sort order.
-y6	Specifies that directory records be exactly aligned at the end of sectors.
-yo	Specifies file layout on disk. This option specifies a text file that has a layout for the files to be placed in the image. The rules for this file are listed below. The order file shall be in ANSI. The order file shall end in a new line. The order file shall have one file per line. Each file shall be specified relative to the root of the image.

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

	Each file shall be specified as a long file name. No short names are allowed. Each file path cannot be longer than MAX_PATH, including volume name. Note that not all files must be listed in the order file. Any files that are not listed in this file shall be ordered as they would be if there was no ordering file. IMPORTANT: The -yo option takes precedence over the -y5 option.
-yw	Opens source files with write sharing.
SourceLocation	Required. Specifies the location of the files that you intend to build into an .iso image.
TargetFile	Specifies the name of the .iso image file.

DiskPart

DiskPart is a text-mode command interpreter in Windows Vista, Windows® XP, and the Windows Server 2003® family. This tool enables you to manage objects (disks, partitions, or volumes) by using scripts or direct input at a command prompt.

DiskPart Commands

Before you can use DiskPart commands on a disk, partition, or volume, you must first list and then select the object to give it focus. When an object has focus, any DiskPart commands that you type act on that object.

You can list the available objects and determine an object's number or drive letter by using the list disk, list volume, and list partition commands. The list disk and list volume commands display all disks and volumes on the computer. However, the list partition command displays only partitions on the disk that have focus. When you use the list commands, an asterisk (*) appears next to the object with focus. You select an object by its number or drive letter, such as disk 0, partition 1, volume 3, or volume C.

When you select an object, the focus remains on that object until you select a different object. For example, if the focus is set on disk 0, and you select volume 8 as on disk 2, the focus shifts from disk 0 to disk 2, volume 8. Some commands automatically change the focus. For example, when you create a new partition, the focus automatically changes to the new partition.

You can give focus only to a partition on the selected disk. When a partition has focus, the related volume (if any) also has focus. When a volume has focus, the related disk and partition also have focus if the volume maps to a single specific partition. If this is not the case, then focus on the disk and partition is lost.

IMPORTANT: When using the DiskPart command as a part of a script, it is recommended that you complete all of the DiskPart operations together as part of a single DiskPart script. You can run consecutive DiskPart scripts, but you must allow at least 15 seconds between each script for a complete shutdown of the previous execution before running the DiskPart command again in successive scripts. Otherwise, the successive scripts might fail. You can add a pause between consecutive DiskPart scripts by adding the timeout /t 15 command to your batch file along with your DiskPart scripts.

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

DiskPart Syntax and Parameters

This table identifies the syntax and parameters of the DiskPart commands.

Command	Syntax	Description
active	active	<p>On basic disks, marks the partition with focus as active. This informs the basic input/output system (BIOS) or Extensible Firmware Interface (EFI) that the partition or volume is a valid system partition or system volume. Only partitions can be marked as "active."</p> <p>IMPORTANT: DiskPart verifies that only the partition is capable of containing an operating system's startup files. DiskPart does not check the contents of the partition. If you mark a partition as "active" and it does not contain the operating system's startup files, your computer might not start.</p>
add disk	add disk=n [noerr]	<p>Mirrors the simple volume with focus to the specified disk. This command is not valid on Windows Vista.</p> <p>n Specifies the disk to contain the mirror. You can mirror only simple volumes. The specified disk must have unallocated space at least as large as the size of the simple volume that you intend to mirror.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
assign	assign [{(letter=d mount=path)}] [noerr]	<p>Assigns a drive letter or mount point to the volume with focus. If no drive letter or mount point is specified, then the next available drive letter is assigned. If the assigned drive letter or mount point is already in use, an error is generated.</p> <p>By using the assign command, you can change the drive letter associated with a removable drive. You cannot assign drive letters to system volumes, boot volumes, or volumes that contain the paging file. You cannot assign a drive letter to an OEM partition or any GPT partition other than a basic data partition.</p> <p>letter=d Specifies the drive letter that you intend to assign to the volume.</p> <p>mount=path Specifies the mount point path that you intend to assign to the volume.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
attributes	attributes volume [{(set clear)}] [{(hidden readonly nodefualtdriveletter shadowcopy)}] [noerr]	<p>volume Displays the attributes of the selected volume.</p> <p>set Sets the specified attribute (hidden, read-only, nodefualtdriveletter, or shadowcopy volume) on the</p>

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

		<p>selected volume.</p> <p>clear Clears the specified attribute (hidden, read-only, nodefaultdriveletter, or shadowcopy volume) from the selected volume.</p> <p>hidden Specifies that the volume is hidden.</p> <p>readonly Specifies that the volume is read-only.</p> <p>nodefaultdriveletter Specifies that the volume does not receive a drive letter by default.</p> <p>shadowcopy Specifies that the volume is a shadow copy volume.</p> <p>noerr For scripting only. When an error is encountered, DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
<p>automount</p>	<p>automount [enable] [disable] [scrub] [noerr]</p>	<p>When enabled (the default), Windows automatically mounts the file system for a new basic volume when it is added to the system, and then assigns a drive letter to the volume. In system area network configurations, disabling automount prevents Windows from automatically mounting or assigning drive letters to any new basic volumes added to the system.</p> <p>enable Enables Windows to automatically mount new basic volumes added to the system and to assign them drive letters.</p> <p>disable Prevents Windows from automatically mounting any new basic volumes that are added to the system.</p> <p>scrub Removes volume mount point directories and registry settings for volumes that are no longer in the system. This prevents volumes that were previously in the system from being automatically mounted and given their former volume mount point(s) when they are added back to the system.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
<p>break disk</p>	<p>break disk=n [nokeep] [noerr]</p>	<p>Applies to dynamic disks only. Breaks the mirrored volume with focus into two simple volumes. One simple volume retains the drive letter and any mount points of the mirrored volume, while the other simple volume receives the focus so that you can assign it a drive letter.</p> <p>By default, the contents of both halves of the mirror are retained; each half becomes a simple volume. If you use the nokeep parameter, only one-half of the mirror is retained as a simple volume, while the other half is deleted and converted to free space. Neither volume receives the focus.</p> <p>n</p>

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

		<p>Specifies the disk that contains the mirrored volume.</p> <p>nokeep Specifies that only one of the mirrored volumes is retained; the other simple volume is deleted and converted to free space. Neither volume receives the focus.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
clean	clean [all]	<p>Removes any and all partition or volume formatting from the disk with focus. On master boot record (MBR) disks, only the MBR partitioning information and hidden sector information are overwritten. On GUID partition table (GPT) disks, the GPT partitioning information, including the Protective MBR, is overwritten; there is no hidden sector information.</p> <p>all Specifies that each and every sector on the disk is zeroed, which completely deletes all data contained on the disk.</p>
convert basic	convert basic [noerr]	<p>Converts an empty dynamic disk into a basic disk.</p> <p>IMPORTANT: The disk must be empty to convert it to a basic disk. Back up your data, and then delete all partitions or volumes before converting the disk.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
convert dynamic	convert dynamic [noerr]	<p>Converts a basic disk into a dynamic disk. Any existing partitions on the disk become simple volumes.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
convert gpt	convert gpt [noerr]	<p>On Itanium-based computers, converts an empty basic disk with the master boot record (MBR) partition style into a basic disk with the GUID partition table (GPT) partition style.</p> <p>IMPORTANT: The disk must be empty to convert it to a GPT disk. Back up your data and then delete all partitions or volumes before converting the disk.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
convert mbr	convert mbr [noerr]	<p>On Itanium-based computers, converts an empty basic disk with the GUID Partition Table (GPT) partition style to a basic disk with the master boot record (MBR) partition style.</p>

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

		<p>IMPORTANT: The disk must be empty to convert it to an MBR disk. Back up your data and then delete all partitions or volumes before converting the disk.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
create partition efi	create partition efi [size=n] [offset=n] [noerr]	<p>On Itanium-based computers, creates an Extensible Firmware Interface (EFI) system partition on a GUID Partition Table (GPT) disk. After the partition has been created, the focus is given to the new partition.</p> <p>size=n Specifies the size of the partition in megabytes (MB). If no size is given, the partition continues until there is no more free space in the current region.</p> <p>offset=n Specifies the byte offset at which to create the partition. If no offset is given, the partition is placed in the first disk extent that is large enough to hold it.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
create partition extended	create partition extended [size=n] [offset=n] [noerr]	<p>Creates an extended partition on the current drive. After the partition has been created, the focus automatically shifts to the new partition. Only one extended partition can be created per disk. This command fails if you attempt to create an extended partition within another extended partition. You must create an extended partition before you can create logical drives.</p> <p>size=n Specifies the size of the extended partition in megabytes (MB). If no size is given, then the partition continues until there is no more free space in the region. The size is cylinder snapped; that is, the size is rounded to the closest cylinder boundary. For example, if you specify a size of 500 MB, the partition size rounds up to 504 MB.</p> <p>offset=n Applies to master boot record (MBR) disks only. Specifies the byte offset at which to create the extended partition. If no offset is given, the partition starts at the beginning of the first free space on the disk. The offset is cylinder snapped; that is, the offset is rounded to the closest cylinder boundary. For example, if you specify an offset that is 27 MB and the cylinder size is 8 MB, the offset is rounded to the 24-MB boundary.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
create partition logical	create partition logical [size=n] [offset=n] [noerr]	<p>Creates a logical drive in the extended partition. After the partition has been created, the focus automatically shifts to the new logical drive.</p>

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

		<p>size=n The size of the logical drive in megabytes (MB). If no size is given, then the partition continues until there is no more free space in the current region.</p> <p>offset=n Applies to master boot record (MBR) disks only. Specifies the byte offset at which to create the logical drive. The offset is cylinder snapped; that is, the offset rounds up to completely fill whatever cylinder size is used. If no offset is given, then the partition is placed in the first disk extent that is large enough to hold it. The partition is at least as long in bytes as the number specified by size=n. If you specify a size for the logical drive, it must be smaller than the extended partition.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
create partition msr	create partition msr [size=n] [offset=n] [noerr]	<p>On Itanium-based computers, creates a Microsoft reserved (MSR) partition on a GUID Partition Table (GPT) disk.</p> <p>CAUTION: Be very careful when using the create partition msr command. GPT disks require a specific partition layout, and so creating Microsoft Reserved Partitions can cause the disk to become unreadable. On GPT disks that are used to start Windows XP 64-Bit Edition or the 64-bit versions of the Windows Server 2003 family, the EFI system partition is the first partition on the disk, followed by the Microsoft Reserved Partition. GPT disks used only for data storage do not have an EFI system partition; the Microsoft Reserved Partition is the first partition. Windows XP and the Windows Server 2003 family do not mount Microsoft Reserved Partitions. You cannot store data on them and you cannot delete them.</p> <p>size=n Specifies the size of the partition in megabytes (MB). The partition is at least as long in bytes as the number specified by size=n. If no size is given, the partition continues until there is no more free space in the current region.</p> <p>offset=n Specifies the byte offset at which to create the partition. The partition starts at the byte offset specified by offset=n. It is sector-snapped; that is, the offset rounds up to completely fill whatever sector size is used. If no offset is given, then the partition is placed in the first disk extent that is large enough to hold it.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
create partition primary	create partition primary [size=n] [offset=n] [align=n][noerr] [ID={byte GUID}]	<p>Creates a primary partition on the current basic disk. After you create the partition, the focus automatically shifts to the new partition. The partition does not receive a drive letter; you must use the assign command to assign a</p>

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

		<p>drive letter to the partition.</p> <p>size=n Specifies the size of the partition in megabytes (MB). If no size is given, the partition continues until there is no more unallocated space in the current region. The size is cylinder snapped; that is, the size rounds to the closest cylinder boundary. For example, if you specify a size of 500 MB, the partition size rounds up to 504 MB.</p> <p>offset=n Specifies the byte offset at which to create the partition. If no offset is given, the partition starts at the beginning of the first free space on the disk. For master boot record (MBR) disks, the offset is cylinder snapped; that is, the offset rounds to the closest cylinder boundary. For example, if you specify an offset that is 27 MB and the cylinder size is 8 MB, the offset rounds to the 24-MB boundary.</p> <p>ID={byte GUID} Intended for OEM use only. CAUTION: Creating partitions with this parameter might cause your computer to crash or be unable to start. Unless you are an OEM or an IT professional experienced with GPT disks, do not create partitions on GPT disks by using the ID=byte GUID parameter. Instead, always use the create partition efi command to create EFI system partitions, the create partition msr command to create Microsoft Reserved Partitions, and the create partition primary command (without the ID=byte GUID parameter) to create primary partitions on GPT disks.</p> <p>align= n Typically used with hardware RAID Logical Unit Number (LUN) arrays to improve performance when the logical units (LUs) are not cylinder aligned. Aligns a primary partition that is not cylinder aligned at the beginning of a disk and rounds the offset to the closest alignment boundary, where n is the number of kilobytes (KB) from the beginning of the disk to the closest alignment boundary. The align= n command fails if the primary partition is not at the beginning of the disk. If used with offset= n, the offset is within the first usable cylinder on the disk.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p> <p>Comments To partition a GPT disk with an OEM partition generate an OEM-GUID: Use this command to create an EFI partition: create partition efi size=n Use this command to create a primary partition: create partition primary size=n ID={byte GUID} Use this command to create an MSR partition: create partition MSR size=n</p> <p>IMPORTANT: You must create the OEM partition between the EFI and MSR partitions. Never create the LDM metadata or LDM data partitions explicitly as partitions. Instead, convert the disk to dynamic.</p>
--	--	---

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

		<p>For master boot record (MBR) disks, you can specify a partition type byte, in hexadecimal form, for the partition. If you do not specify a partition type byte on an MBR disk, the create partition primary command creates a partition of type 0x6. Any partition type byte can be specified with the ID=byte GUID parameter. DiskPart does not verify the validity of the partition type nor does it verify the ID parameter.</p> <p>For GPT disks, you can specify a partition type GUID for the partition that you create.</p> <p>EFI system partition: c12a7328-f81f-11d2-ba4b-00a0c93ec93b</p> <p>Microsoft reserved partition: e3c9e316-0b5c-4db8-817d-f92df00215ae</p> <p>Basic data partition: ebd0a0a2-b9e5-4433-87c0-68b6b72699c7</p> <p>LDM metadata partition on a dynamic disk: 5808c8aa-7e8f-42e0-85d2-e1e90434cfb3</p> <p>LDM data partition on a dynamic disk: af9b60a0-1431-4f62-bc68-3311714a69ad</p> <p>If you do not specify a partition type GUID, the create partition primary command creates a basic data partition. Any partition type can be specified with the ID=byte GUID parameter. DiskPart does not verify the validity of the partition GUID nor does it verify the ID parameter.</p>
<p>create volume raid</p>	<p>create volume raid [size=n] [disk=n,n,n[,...]] [noerr]</p>	<p>Creates a RAID-5 volume on three or more specified dynamic disks. After you create the volume, the focus automatically shifts to the new volume. This command is not valid on Windows Vista.</p> <p>size=n</p> <p>Specifies the amount of disk space, in megabytes (MB), that the volume occupies on each disk. If no size is given, the largest possible RAID-5 volume is created. The disk with the smallest available contiguous free space determines the size for the RAID-5 volume and the same amount of space is allocated from each disk. The actual amount of usable disk space in the RAID-5 volume is less than the combined amount of disk space because some of the disk space is required for parity.</p> <p>disk=n,n,n[,...]</p> <p>Specifies the dynamic disks on which to create the volume. You need at least three dynamic disks in order to create a RAID-5 volume. An amount of space equal to size=n is allocated on each disk.</p> <p>noerr</p> <p>For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
<p>create volume simple</p>	<p>create volume simple [size=n] [disk=n] [noerr]</p>	<p>Creates a simple volume. After you create the volume, the focus automatically shifts to the new volume. Dynamic disks are not supported on Windows Vista Starter, Windows Vista Home Basic, Windows Vista Home N, and Windows Vista Business operating systems.</p> <p>size=n</p> <p>Specifies the size of the volume in megabytes (MB). If no size is given, the new volume takes up the remaining free</p>

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

		<p>space on the disk.</p> <p>disk=<i>n</i> Specifies the dynamic disk on which to create the volume. If no disk is given, the current disk is used.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
create volume stripe	create volume stripe [size= <i>n</i>] [disk= <i>n</i> , <i>n</i> [, <i>n</i> ,...]] [noerr]	<p>Creates a striped volume by using two or more specified dynamic disks. After you create the volume, the focus automatically shifts to the new volume. Dynamic disks are not supported on Windows Vista Starter, Windows Vista Home Basic, Windows Vista Home N, and Windows Vista Business operating systems.</p> <p>size=<i>n</i> Specifies the amount of disk space, in megabytes (MB), that the volume occupies on each disk. If no size is given, the new volume takes up the remaining free space on the smallest disk and an equal amount of space on each subsequent disk.</p> <p>disk=<i>n</i>,<i>n</i>[,<i>n</i>,...] Specifies the dynamic disks on which to create the volume. You need at least two dynamic disks to create a striped volume. An amount of space equal to size=<i>n</i> is allocated on each disk.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
delete disk	delete disk [noerr] [override]	<p>Deletes a missing dynamic disk from the disk list.</p> <p>noerr For scripting only. When an error is encountered, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p> <p>override Enables DiskPart to delete all simple volumes on the disk. If the disk contains half of a mirrored volume, the half of the mirror on the disk is deleted. The delete disk override command fails if the disk is a member of a RAID-5 volume.</p>
delete partition	delete partition [noerr] [override]	<p>On a basic disk, deletes the partition with focus. You cannot delete the system partition, boot partition, or any partition that contains the active paging file or crash dump (memory dump).</p> <p>CAUTION: Deleting a partition on a dynamic disk can delete all dynamic volumes on the disk, thus destroying any data and leaving the disk in a corrupted state. To delete a dynamic volume, always use the delete volume command instead.</p> <p>You can delete partitions from dynamic disks, but you must not create them. For example, it is possible to delete an unrecognized GUID Partition Table (GPT) partition on a dynamic GPT disk. However, deleting such a partition</p>

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

		<p>does not cause the resulting free space to become available. This command is intended to enable space reclamation on a corrupted, offline dynamic disk in an emergency situation where the clean command cannot be used.</p> <p>noerr For scripting only. When an error occurs, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p> <p>override Enables DiskPart to delete any partition regardless of type. Normally, DiskPart enables you to delete only known data partitions.</p>
delete volume	delete volume [noerr]	<p>Deletes the selected volume. You cannot delete the system volume, boot volume, or any volume that contains the active paging file or crash dump (memory dump).</p> <p>noerr For scripting only. When an error occurs, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
detail disk	detail disk	Displays the properties of the selected disk and the volumes on that disk.
detail partition	detail partition	Displays the properties of the selected partition.
detail volume	detail volume	Displays the disks on which the current volume resides.
exit	exit	Exits the DiskPart command interpreter.
extend	extend [size=n] [disk=n] [noerr]	<p>Extends the volume with focus into the next contiguous unallocated space. For basic volumes, the unallocated space must be on the same disk as, and must follow (have a higher sector offset number than) the partition with focus. A dynamic, simple, or spanned volume can be extended to any empty space on any dynamic disk. By using this command, you can extend an existing volume into newly created space.</p> <p>If the partition was previously formatted with the NTFS file system, the file system is automatically extended to occupy the larger partition. No data loss occurs. If the partition was previously formatted with any file system format other than NTFS, the command fails with no change to the partition.</p> <p>You cannot extend the current system or boot partitions.</p> <p>size=n Specifies the amount of space, in megabytes (MB), to add to the current partition. If you do not specify a size, the disk is extended to take up all of the next contiguous unallocated space.</p> <p>disk=n Specifies the dynamic disk on which to extend the volume. An amount of space equal to size=n is allocated on the disk. If no disk is specified, the volume is extended</p>

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

		<p>on the current disk.</p> <p>noerr For scripting only. When an error occurs, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
filesystems	filesystems	<p>Displays current and supported file systems on the volume. A volume must be selected for this operation to succeed.</p>
format	<p>format [[FS=<FS>] [REVISION=<X.XX>] RECOMMENDED]] [LABEL=<"label">] [UNIT=<N>] [QUICK] [COMPRESS] [OVERRIDE] [NOWAIT] [NOERR]</p>	<p>Formats the volume or partition. A volume must be selected for this operation to succeed.</p> <p>FS=<FS> Specifies the type of file system. If no file system is given, the default file system displayed by the FILESYSTEMS command is used.</p> <p>REVISION = <X.XX> Specifies the file system revision (if applicable).</p> <p>RECOMMENDED If specified, use the recommended file system and revision instead of the default if a recommendation exists. The recommended file system (if one exists) is displayed by the FILESYSTEMS command.</p> <p>LABEL=<"label"> Specifies the volume label.</p> <p>UNIT=<N> Overrides the default allocation unit size. Default settings are strongly recommended for general use. The default allocation unit size for a particular file system is displayed by the FILESYSTEMS command.</p> <p>NTFS compression is not supported for allocation unit sizes above 4096.</p> <p>QUICK Performs a quick format.</p> <p>COMPRESS NTFS only: Files created on the new volume will be compressed by default.</p> <p>OVERRIDE Forces the volume to dismount first if necessary. All opened handles to the volume would no longer be valid.</p> <p>NOWAIT Forces the command to return immediately while the format process is still in progress. If NOWAIT is not specified, DiskPart will display format progress in percentage.</p> <p>NOERR For scripting only. When an error is encountered, DiskPart continues to process commands as if the error did not occur. Without the NOERR parameter, an error causes DiskPart to exit with an error code.</p> <p>Examples: FORMAT FS=NTFS LABEL="New Volume" QUICK COMPRESS FORMAT RECOMMENDED OVERRIDE</p>
gpt attributes	gpt attributes= n	<p>On basic GPT disks, assigns the GPT attribute(s) to the partition with focus. GPT partition attributes give additional information about the use of the partition. Some</p>

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

		<p>attributes are specific to the partition type GUID.</p> <p>IMPORTANT: Changing the GPT attributes might cause your basic data volumes to be unmountable or fail to be assigned drive letters. Unless you are an OEM or an IT professional experienced with GPT disks, do not change GPT attributes.</p> <p>n</p> <p>The hexadecimal that pertains to the attribute that you intend to apply to the partition with focus. The GPT attribute field is a 64-bit field that contains two subfields. The higher field is interpreted only in the context of the partition ID, while the lower field is common to all partition IDs.</p> <p>All partitions have the following attribute: 0x0000000000000001 marks the partition as required. This indicates to all disk management utilities that the partition must not be deleted. The EFI system partition contains only those binaries necessary to start the operating system. This makes it easy for OEM- or operating system-specific binaries to be placed in other partitions.</p> <p>For basic data partitions, the following attribute is defined: 0x8000000000000000 prevents the partition from having a drive letter automatically assigned. By default, each partition is assigned a new drive letter. Setting this attribute ensures that, when a disk is moved to a new computer, a new drive letter will not be automatically generated. Instead, the user can manually assign drive letters.</p> <p>NOTE: Other attributes can be added at any time.</p>
help	help	Displays a list of the available commands.
import	import [noerr]	<p>Imports a foreign disk group into the local computer's disk group. The import command imports every disk that is in the same group as the disk that has focus.</p> <p>noerr</p> <p>For scripting only. When an error occurs, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
inactive	inactive	<p>On basic master boot record (MBR) disks, marks the system partition or boot partition with focus as inactive. The computer starts from the next option specified in the BIOS such as the CD-ROM drive or a Pre-Boot eXecution Environment (PXE)-based boot environment (such as Remote Installation Services (RIS)) when you restart the computer.</p> <p>CAUTION: Your computer might not start without an active partition. Do not mark a system or boot partition as inactive unless you are an experienced user with a thorough understanding of the Windows Server 2003 family.</p> <p>If you are unable to start your computer after marking the system or boot partition as inactive, insert the Setup CD in the CD-ROM drive, restart the computer, and then repair the partition by using the Fixmbr and Fixboot commands in the Recovery Console.</p>

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

list disk	list disk	Displays a list of disks and information about them, such as their size, amount of available free space, whether the disk is a basic or dynamic disk, and whether the disk uses the master boot record (MBR) or GUID partition table (GPT) partition style. The disk marked with an asterisk (*) has focus.
list partition	list partition	Displays the partitions listed in the partition table of the current disk. On dynamic disks, these partitions may not correspond to the dynamic volumes on the disk. This discrepancy occurs because dynamic disks contain entries in the partition table for the system volume or boot volume (if present on the disk). Dynamic disks also contain a partition that occupies the remainder of the disk and reserves space for use by dynamic volumes.
list volume	list volume	Displays a list of basic and dynamic volumes on all disks.
online	online [noerr]	Brings an offline disk or volume with focus online. Resynchronizes the mirrored or RAID-5 volume with focus. noerr For scripting only. When an error occurs, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.
rem	rem	Provides a way to add comments to a script. For example: rem These commands set up 3 drives. create partition primary size=2048 assign d: create partition extend create partition logical size=2048 assign e: create partition logical assign f:
remove	remove [{letter=d mount=path[[all]]] [dismount][noerr]	Removes a drive letter or mount point from the volume with focus. If the all parameter is used, all current drive letters and mount points are removed. If you do not specify a drive letter or mount point, then DiskPart removes the first drive letter or mount point that it encounters. You can use the remove command to change the drive letter associated with a removable drive. You cannot remove the drive letters on system, boot, or paging volumes. In addition, you cannot remove the drive letter for an OEM partition, any GPT partition with an unrecognized GUID, or any of the special, non-data, GPT partitions such as the EFI system partition. letter=d Specifies the drive letter to remove. mount=path Specifies the mount point path to remove. all Removes all current drive letters and mount points. dismount Dismounts the basic volume, when all drive letters and mount points have been removed from the volume, and

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

		<p>takes the basic volume offline, making it unmountable. If other processes are using the volume, DiskPart closes any open handles before dismounting the volume. You can make the volume mountable by assigning it a drive letter or by creating a mount point path to the volume. Dismount will fail if used on a volume that has any remaining drive letters or mount points. For scripting, using removeall dismount is recommended.</p> <p>noerr For scripting only. When an error occurs, specifies that DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
repair disk	repair disk= <i>n</i> [noerr]	<p>Repairs the RAID-5 volume with focus by replacing the failed RAID-5 member with the specified dynamic disk. The specified dynamic disk must have free space greater than or equal to the total size of the failed RAID-5 member.</p> <p><i>n</i> Specifies the dynamic disk that replaces the failed RAID-5 member. The specified disk must have free space equal to or larger than the total size of the failed RAID-5 member.</p> <p>noerr For scripting only. When an error occurs, DiskPart continues to process commands as if the error did not occur. Without the noerr parameter, an error causes DiskPart to exit with an error code.</p>
rescan	rescan	<p>Locates new disks that might have been added to the computer.</p>
retain	retain	<p>Prepares an existing dynamic simple volume to use as a boot or system volume.</p> <p>On an x86-based computer, creates a partition entry in the master boot record (MBR) on the dynamic simple volume with focus. To create an MBR partition, the dynamic simple volume must start at a cylinder aligned offset and be an integral number of cylinders in size.</p> <p>On an Itanium-based computer, creates a partition entry in the GUID partition table (GPT) on the dynamic simple volume with focus.</p> <p>NOTE: The retain command is intended for use only during unattended Setup or by OEMs.</p>
select disk	select disk= <i>n</i>]	<p>Selects the specified disk and shifts the focus to it.</p> <p><i>n</i> Specifies the disk number of the disk to receive focus. If you do not specify a disk number, the select command lists the disk that currently has the focus. You can view the numbers for all disks on the computer by using the list disk command.</p>
select partition	select partition= <i>[n d]</i>]	<p>Selects the specified partition and gives it focus. If you do not specify a partition, the select command lists the current partition with focus. You can view the numbers of all partitions on the current disk by using the list partition command.</p>

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

		<p>n Specifies the number of the partition to receive the focus.</p> <p>d Specifies the drive letter or mount point path of the partition to receive the focus.</p>
select volume	select volume={n d}	<p>Selects the specified volume and shifts the focus to it. If you do not specify a volume, the select command lists the current volume with focus. You can specify the volume by number, drive letter, or mount point path. On a basic disk, selecting a volume also gives the corresponding partition focus. You can view the numbers of all volumes on the computer by using the list volume command.</p> <p>n Specifies the number of the volume to receive the focus.</p> <p>d Specifies the drive letter or mount point path of the volume to receive the focus.</p>
setid	set id=<BYTE> <GUID> [OVERRIDE] [NOERR]	<p>Change the partition type. Intended for Original Equipment Manufacturer (OEM) use only. A partition must be selected for this operation to succeed.</p> <p>ID=<BYTE> <GUID></p> <p>Specifies the new partition type. For master boot record (MBR) disks, you can specify a partition type byte, in hexadecimal form, for the partition. Any partition type byte can be specified with this parameter except for type 42 (LDM partition).</p> <p>For GUID partition table (GPT) disks you can specify a partition type GUID for the partition.</p> <p>OVERRIDE Enables DiskPart to force the volume to dismount first if necessary. All opened handles to the volume will become invalid.</p> <p>NOERR For scripting only. When an error is encountered, DiskPart continues to process commands as if the error did not occur. Without the NOERR parameter, an error causes DiskPart to exit with an error code.</p> <p>Caution: Changing partition type fields with this parameter might cause your computer to fail or be unable to start up. Unless you are an OEM or an IT professional experienced with GPT disks, do not change partition type fields on GPT disks using this parameter. Instead, always use the CREATE PARTITION EFI command to create EFI System partitions, the CREATE PARTITION MSR command to create Microsoft Reserved partitions, and the CREATE PARTITION PRIMARY command without the ID parameter to create primary partitions on GPT disks. This command does not work on dynamic disks nor on Microsoft Reserved partitions.</p> <p>Example: SET ID=07 OVERRIDE SET ID=ebd0a0a2-b9e5-4433-87c0-68b6b72699c7</p>
shrink	shrink [DESIRED=<N>] [MINIMUM=<N>] [NOWAIT] [NOERR] shrink QUERYMAX [NOERR]	<p>Reduces the size of the volume with focus by the specified amount. Makes free disk space available from unused space at the end of the volume. A volume must be selected for this operation to succeed.</p>

Windows PE
Windows Preinstallation Environment
(Microsoft Corporation)

		<p>DESIRED=<N> Specifies the desired amount of space in megabytes (MB) to reduce the size of the volume by. If a desired amount is not specified, the volume will be reduced by the maximum amount of free space available on the volume.</p> <p>MINIMUM=<N> Specifies the minimum amount of space in MB to reduce the size of the volume by.</p> <p>QUERYMAX Returns the maximum number of bytes that the volume can be reduced by (the free space available on the volume). This value may change if applications are currently accessing the volume.</p> <p>NOWAIT Forces the command to return immediately while the shrink process is still in progress.</p> <p>NOERR For scripting only. When an error is encountered, DiskPart continues to process commands as if the error did not occur. Without the NOERR parameter, an error causes DiskPart to exit with an error code.</p> <p>If a MINIMUM amount is not specified, the volume will be reduced by either the DESIRED amount (if specified), or by the maximum amount of free space available on the volume. If a MINIMUM amount is specified but not enough free space is available, the command will fail.</p> <p>This command works on basic volumes, and on simple or spanned dynamic volumes. You can reduce the size of a volume only if it is formatted using the NTFS file system or if it does not have a file system.</p> <p>Examples: SHRINK DESIRED=500 MINIMUM=250 SHRINK QUERYMAX</p>
--	--	---

DiskPart Scripting

By using the DiskPart Command-Line Options command-line tool, you can create scripts to automate disk-related tasks, such as creating volumes or converting disks to dynamic disks. Scripting these tasks is useful if you deploy Windows by using unattended Setup or the Sysprep tool, which do not support creating volumes other than the boot volume.

To start a DiskPart script, at the command prompt, type:

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

where scriptname is the name of the text file that contains your script.

To redirect DiskPart's scripting output to a file, type:

```
diskpart /s scriptname.txt > logfile.txt
```

where logfile is the name of the text file where DiskPart writes its output.

When DiskPart starts, the DiskPart version and computer name display at the command prompt. By default, if DiskPart encounters an error while attempting to perform a scripted task, DiskPart stops

Windows PE

Windows Preinstallation Environment (Microsoft Corporation)

processing the script and displays an error code (unless you specified the noerr parameter). However, DiskPart always returns errors when it encounters syntax errors, regardless of whether you used the noerr parameter. The noerr parameter enables you to perform useful tasks such as using a single script to delete all partitions on all disks regardless of the total number of disks.

The following table lists the DiskPart error codes.

Error	Description
0	No errors occurred. The entire script ran without failure.
1	A fatal exception occurred. There might be a serious problem.
2	The parameters specified for a DiskPart command are incorrect.
3	DiskPart was unable to open the specified script or output file.
4	One of the services that DiskPart uses returned a failure.
5	A command syntax error occurred. The script failed because an object was improperly selected or was invalid for use with that command.

Drvload

The Drvload tool adds out-of-box drivers to a booted Windows PE image. It takes one or more driver .inf files as inputs. To add a driver to an offline Windows PE image, use the peimg tool.

If the driver .inf file requires a reboot, Windows PE will ignore the request. If the driver .sys file requires a reboot, then the driver cannot be added with Drvload.

Drvload Command-Line Options

The following command-line options are available for Drvload.

```
drvload.exe inf_path [,inf_path [...]] [/?]
```

Option	Description
/?	Displays usage information.
inf_path	Specifies the path to the driver .inf file. The path can contain environment variables.

If any drivers failed to install, then Drvload will return 0 if successful or a non-zero status (%errorlevel%).

PEimg

PEimg.exe is a command-line tool for creating and modifying Windows PE 2.0 images offline.

PEimg Command-Line Options

The following command-line options are available for PEimg.exe.

Windows PE

Windows Preinstallation Environment (Microsoft Corporation)

```
peimg {/import | /inf | /install | /lang | /scratchspace | /targetpath
| /timezone | /list | /prep | /uninstall } <%WINDIR%> [/verbose] [/quiet]
```

Option	Description
/?	Displays command-line help.
<%WINDIR%>	Specifies the path to the Windows directory within the base Windows PE image. The image must first be applied or mounted to a local directory by using ImageX. Path can also be written as /image=<path>.
/import=<path>	Imports a Windows PE package from a .cab file or directory structure. The package will be made available for installation. After importing a package, you must install the package by using the /install option. Winpe.wim provides several Windows PE packages. For example, peimg /import=c:\feature.cab c:\winpe_x86\mount\windows
/inf=<inf>	Installs an .inf package (typically a driver) to a Windows PE image, where <path> is the path to the .inf file or package. You can perform this on an image that was previously prepared with the /prep option. For, example, peimg /inf=c:\mydevice.inf c:\winpe_x86\mount\windows
/install=<pkg>	Installs a package, where <pkg> denotes the package name. A list of available packages and their names can be obtained by using the /list option. Wildcards can be used when specifying 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.
/lang=<culture> [/layerdriver=<ID>]	Sets the locale and the user interface language of a Windows PE image, where <culture> is the culture name of the specified language. A language pack for the specified language must already be installed. For the ja-JP or ko-KR cultures, the /layerdriver option can be used to specify which keyboard layer driver should be installed. For a complete list of <culture> and <ID> values, see Supported Language Packs and Default Settings. You can perform this command on a Windows PE image that was previously prepared with the /prep option. For example, peimg /lang=en-us c:\winpe_x86\mount\windows
/scratchspace =<size>	Sets the available scratch space, in megabytes. Valid values are 32, 64, 128, 256 and 512.
/targetpath=X:\	For hard disk boot scenarios, sets the location of the Windows PE image on the disk. This path must be equal to or a subdirectory of X:\ (this is the default). You can perform this only on a Windows PE image that was previously prepared with the /prep option.

Windows PE

Windows Preinstallation Environment (Microsoft Corporation)

/timezone =<culture or name>	Sets the default time zone of a Windows PE image. The time zone can be specified by culture name or time zone name. For a complete list of supported <culture> and <name> values, see, Supported Language Packs and Default Settings. You can perform this only on a Windows PE image that was previously prepared with the /prep option.
/list	Displays a list of packages in the current image. In the INS column, a plus sign (+) denotes installed packages and a minus sign (-) denotes not installed. The /verbose option will display the full package name. For example, peimg /list c:\winpe_x86\mount\windows
/prep [/f]	Optimizes the Windows PE image for size. This operation cannot be reverted, and after the /prep option is issued, the /install, /uninstall, /import, and /list options will not function. The /lang and /inf options will continue to function. The tool will prompt the user to confirm the option. To suppress this prompt for scripting, specify the /f option. For example, peimg /prep c:\winpe_x86\mount\windows
/uninstall=<pkg>	Uninstalls a package, where <pkg> is the package name. A list of available packages and their names can be obtained, by using the /list option. Wildcards can be used when specifying a package name; any packages with matching names will be installed. For example, peimg /uninstall=WinPE-HTA-Package c:\winpe_x86\mount\windows -OR- peimg /uninstall=*HTA* c:\winpe_x86\mount\windows where wildcards denote any component with HTA in the package name.

The following options can also be set.

Flag	Description
/verbose	Displays full package identities. For example, peimg /list c:\winpe_x86\mount\windows /verbose
/quiet	Suppresses the progress bar. For example, peimg /prep c:\winpe_x86\mount\windows /quiet

Wpeinit

Wpeinit is a command-line tool that initializes Windows PE each time it boots. When Windows PE starts, Wimpeshl.exe executes Startnet.cmd, which launches Wpeinit.exe. Wpeinit.exe specifically installs PnP devices, processes Unattend.xml settings, and loads network resources.

Wpeinit replaces the initialization function previously supported in Factory.exe -winpe. Wpeinit outputs a log messages to c:\Windows\system32\wpeinit.log.

Windows PE

Windows Preinstallation Environment (Microsoft Corporation)

Wpeinit Command-Line Options

You can manually execute Wpeinit from a Windows PE Command Prompt window to process a custom answer file. Wpeinit.exe accepts one option called -unattend. You can specify the parameter in one of four ways.

```
wpeinit -unattend=<path\unattend>
wpeinit -unattend:<path\unattend>
wpeinit /unattend=<path\unattend>
wpeinit /unattend:<path\unattend>
```

Windows PE Answer Files

You can use the following answer files when you build a customized version of Windows Preinstallation Environment or when you preinstall Windows with Windows PE.

Winpeshl.ini Files

Winpeshl.ini controls whether a customized shell is loaded in Windows PE instead of the default Command Prompt window. To load a customized shell, create a file named Winpeshl.ini and place it in %SYSTEMROOT%\System32 of your customized Windows PE image. The .ini file must have the following section and entry.

```
[LaunchApp]
AppPath = %SYSTEMDRIVE%\myshell.exe
[LaunchApps]
%SYSTEMDRIVE%\mydir\application1.exe, -option1 -option2
application2.exe, -option1 -option2
```

Set the AppPath entry to the path to your shell application. The path can be either fully qualified, or you can use environment variables, such as %SYSTEMROOT%\System32\Myshell.exe. The AppPath entry does not support command-line options.

Use the [LaunchApps] section to run applications with command-line options. The applications run in the order listed; separate the name of the application from its options with a comma.

When you exit the Command Prompt window or your customized shell application, Windows PE restarts.

IMPORTANT: Do not edit the value of CmdLine in the HKEY_LOCAL_MACHINE\SYSTEM\Setup registry key of Setupreg.hiv to start your shell application. The value of CmdLine calls Winpeshl.exe, which performs a number of functions, including calling the default Command Prompt window in Windows PE or your customized shell application.