Learn

Discover ⌄ Product documentation ⌄ Development languages ⌄ Topics ⌄

Sign in

Windows Release health Windows client ⌄ Application developers ⌄ Hardware developers ⌄ Windows Server Windows for IoT Windows Insider Program More ⌄

🗎 Download PDF

Learn / Windows / Windows Drivers /

BCDEdit /set

Article • 11/23/2022 • 6 contributors

🗘 Feedback

In this article

Alternatives to BCDEdit

Syntax

Parameters

Use the command line help to view options

Show 18 more

The **BCDEdit /set** command sets a boot entry option value in the Windows boot configuration data store (BCD). Use the **BCDEdit /set** command to configure specific boot entry elements, such as kernel debugger settings, memory options, or options that enable test-signed kernel-mode code or load alternate hardware abstraction layer (HAL) and kernel files. To remove a boot entry option, use the BCDEdit /deletevalue command.

> ⊗ **Caution**
>
> Administrative privileges are required to use BCDEdit to modify BCD. Changing some boot entry options using the **BCDEdit /set** command could render your computer inoperable. As an alternative, use the Startup settings or the System Configuration utility (MSConfig.exe) to change boot settings.

> ⓘ **Note**
>
> Before setting BCDEdit options you might need to disable or suspend BitLocker and Secure Boot on the computer.

Alternatives to BCDEdit

Settings startup options

> 💡 **Tip**
>
> To avoid the risk associated with using BCDEdit, consider using an alternative method to perform boot configuration discussed in this section.

Startup Settings

Some common boot options such as enabling debugging mode are available in the start up options. In Windows 10, the settings can be accessed in Settings, Update and Security, select Recovery. Under Advanced startup, select Restart Now. When the PC reboots, select Startup options. Then select Troubleshoot > Advanced options > Startup Settings , then select Restart button. When the PC restarts, you will be able to set the available startup options.

System Configuration Utility

Use the System Configuration Utility (MSConfig.exe) instead of BCDEdit when possible. For more information, see How to open MSConfig in Windows 10 .

Syntax

```syntax
bcdedit  /set [{ID}] datatype value
```

Parameters

[{ID}]
The **{ID}** is the GUID that is associated with the boot entry. If you do not specify an **{ID}**, the command modifies the current operating system boot entry. If a boot entry is specified, the GUID associated with the boot entry must be enclosed in braces **{ }**. To view the GUID identifiers for all of the active boot entries, use the **bcdedit /enum** command. The identifier for the current boot entry is **{current}**. For more information about this option, use the following command: **bcdedit /? ID**
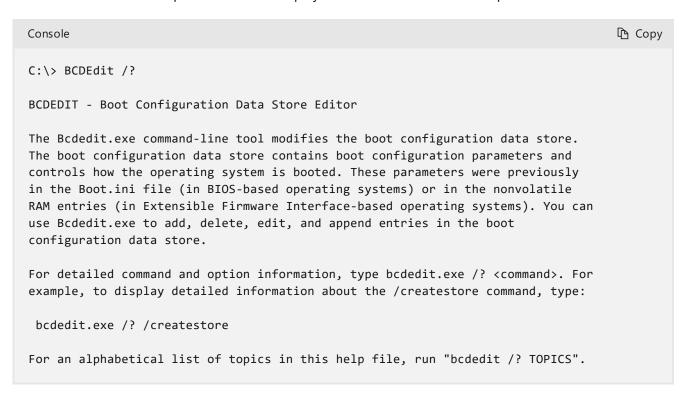
> ⓘ **Note**
>
> If you are using [Windows PowerShell](#), you must use quotes around the boot entry identifier, for example: "**{49916baf-0e08-11db-9af4-000bdbd316a0}**" or "**{current}**".

datatype value

Use the command line help to view options

Use the command line help for BCDEdit to display information available for a specific version of Windows.

```Console
C:\> BCDEdit /?

BCDEDIT - Boot Configuration Data Store Editor

The Bcdedit.exe command-line tool modifies the boot configuration data store.
The boot configuration data store contains boot configuration parameters and
controls how the operating system is booted. These parameters were previously
in the Boot.ini file (in BIOS-based operating systems) or in the nonvolatile
RAM entries (in Extensible Firmware Interface-based operating systems). You can
use Bcdedit.exe to add, delete, edit, and append entries in the boot
configuration data store.

For detailed command and option information, type bcdedit.exe /? <command>. For
example, to display detailed information about the /createstore command, type:

 bcdedit.exe /? /createstore

For an alphabetical list of topics in this help file, run "bcdedit /? TOPICS".
```

The following sections describe some common *datatypes* and their associated *values*.

Boot Settings

bootlog [**yes** | **no**]
Enables the system initialization log. This log is stored in the Ntbtlog.txt file in the %WINDIR% directory. It includes a list of loaded and unloaded drivers in text format.

bootmenupolicy [**Legacy** | **Standard**]
Defines the type of boot menu the system will use. For Windows 10, Windows 8.1, Windows 8 and Windows RT the default is **Standard**. For Windows Server 2012 R2, Windows Server 2012, the default is **Legacy**. When **Legacy** is selected, the Advanced options menu (**F8**) is available. When **Standard** is selected, the boot menu appears but only under certain conditions: for example, if there is a startup failure, if you are booting up from a repair disk or installation media, if you have configured multiple boot entries, or if you manually configured the computer to use Advanced startup. When **Standard** is selected, the **F8** key is ignored during boot. Windows 8 PCs start up quickly so there isn't enough time to press **F8**. For more information, see [Windows Startup Settings (including safe mode)](#) ⧉ .

> ⓘ **Note**

> The option is available starting with Windows 8 and Windows Server 2012. You can also use the **onetimeadvancedoptions** to use the Advanced options (**F8**) menu (**Legacy**) one time on the next boot.

bootstatuspolicy *policy*

Controls the boot status policy. The boot status *policy* can be one of the following:

⬚ Expand table

Boot Status Policy	Description
DisplayAllFailures	Displays all errors if there is a failed boot, failed shutdown, or failed checkpoint. The computer will fail over to the Windows recovery environment on reboot.
IgnoreAllFailures	Ignore errors if there is a failed boot, failed shutdown, or failed checkpoint. The computer will attempt to boot normally after an error occurs.
IgnoreShutdownFailures	Only ignore errors if there is a failed shutdown. If there is a failed shutdown, the computer does not automatically fail over to the Windows recovery environment on reboot. This is the default setting for Windows 8.
IgnoreBootFailures	Only ignore errors if there is a failed boot. If there is a failed boot, the computer does not automatically fail over to the Windows recovery environment on reboot.
IgnoreCheckpointFailures	Only ignore errors if there is a failed checkpoint. If there is a failed checkpoint, the computer does not automatically fail over to the Windows recovery environment on reboot. The option is available starting with Windows 8 and Windows Server 2012.
DisplayShutdownFailures	Displays errors if there is a failed shutdown. If there is a failed shutdown, the computer will fail over to the Windows recovery environment on reboot. Ignores boot failures and failed checkpoints. The option is available starting with Windows 8 and Windows Server 2012.
DisplayBootFailures	Displays errors if there is a failed boot. If there is a failed boot, the computer will fail over to the Windows recovery environment on reboot. Ignores shutdown failures and failed checkpoints. The option is available starting with Windows 8 and Windows Server 2012.
DisplayCheckpointFailures	Displays errors if there is a failed checkpoint. If there is a failed checkpoint, the computer will fail over to the Windows recovery environment on reboot. Ignores boot and shutdown failures. The option is available starting with Windows 8 and Windows Server 2012.

quietboot [**on** | **off**]
Controls the display of a high-resolution bitmap in place of the Windows boot screen display and animation.

> ⓘ **Note**
>
> Do not use the **quietboot** option in Windows 8 as it will prevent the display of bug check data in addition to all boot graphics.

sos [**on** | **off**]
Controls the display of the names of the drivers as they load during the boot process. Use **sos on** to display the names. Use **sos off** to suppress the display.

lastknowngood [**on** | **off**]
Enables boot to last known good configuration.

nocrashautoreboot [**on** | **off**]
Disables automatic restart on crash.

resumeobject (id)
Defines the identifier of the resume object that is associated with this operating system object.

safebootalternateshell [**on** | **off**]
Uses the alternate shell when booted into Safe mode.

winpe [**on** | **off**]
Enables the computer to boot to Windows PE.

onetimeadvancedoptions [**on** | **off**]
Controls whether the system boots to the legacy menu (F8 menu) on the next boot.

syntax   Copy

```
bcdedit /set {current} onetimeadvancedoptions on
```

Display Settings

bootuxdisabled [**on** | **off**]
Disables boot graphics.

graphicsmodedisabled [**on** | **off**] Indicates whether graphics mode is disabled and boot applications must use text mode display.

graphicsresolution
Defines the graphics resolution, 1024x768, 800x600,1024x600, etc.

highestmode [**on** | **off**]
Enables boot applications to use the highest graphical mode exposed by the firmware.

Hardware Abstraction Layer (HAL) & KERNEL

hal *file*
Directs the operating system loader to load an alternate HAL file. The specified file must be located in the %SystemRoot%\system32 directory.

halbreakpoint [**yes** | **no**]
Enables the special hardware abstraction layer (HAL) breakpoint.

kernel *file*
Directs the operating system loader to load an alternate kernel. The specified file must be located in the %SystemRoot%\system32 directory.

useplatformclock [**yes** | **no**]
Forces the use of the platform clock as the system's performance counter.

> ⓘ **Note**
>
> This option should only be used for debugging.

forcelegacyplatform [**yes** | **no**]
Forces the OS to assume the presence of legacy PC devices like CMOS and keyboard controllers.

> ⓘ **Note**
>
> This option should only be used for debugging.

tscsyncpolicy [**Default** | **Legacy** | **Enhanced**]
Controls the times stamp counter synchronization policy. This option should only be used for debugging. Can be Default, Legacy or Enhanced.

Verification Settings

testsigning [**on** | **off**]
Controls whether Windows 10, Windows 8.1, Windows 8, Windows 7, Windows Server 2008, or Windows Vista will load any type of test-signed kernel-mode code. This option is not set by default, which means test-signed kernel-mode drivers on 64-bit versions of Windows 10, Windows 8.1, Windows 8, Windows 7, Windows Server 2008, and Windows Vista will not load by default. After you run the BCDEdit command, restart the computer so that the change takes effect. For more information, see Introduction to Test-Signing

nointegritychecks [**on** | **off**] Disables integrity checks. Cannot be set when secure boot is enabled. This value is ignored by Windows 7 and Windows 8.

disableelamdrivers [**yes** | **no**]
Controls the loading of Early Launch Antimalware (ELAM) drivers. The OS loader removes this entry for security reasons. This option can only be triggered by using the F8 menu. Someone must be physically present (at the computer) to trigger this option.

> ⓘ **Note**
>
> This option should only be used for debugging.

nx [**Optin** |**OptOut** | **AlwaysOn** |**AlwaysOff**]
Enables, disables, and configures Data Execution Prevention (DEP), a set of hardware and software technologies designed to prevent harmful code from running in protected memory locations. For information about DEP settings, see Data Execution Prevention.

⌞⌝ Expand table

DEP Option	Description
Optin	Enables DEP only for operating system components, including the Windows kernel and drivers. Administrators can enable DEP on selected executable files by using the Application Compatibility Toolkit (ACT).
Optout	Enables DEP for the operating system and all processes, including the Windows kernel and drivers. However, administrators can disable DEP on selected executable files by using **System** in **Control Panel**.
AlwaysOn	Enables DEP for the operating system and all processes, including the Windows kernel and drivers. All attempts to disable DEP are ignored.
AlwaysOff	Disables DEP. Attempts to enable DEP selectively are ignored. On Windows Vista, this parameter also disables Physical Address Extension (PAE). This parameter does not disable PAE on Windows Server 2008.

Processor Settings

groupsize *maxsize*
Sets the maximum number of logical processors in a single processor group, where *maxsize* is any power of 2 between 1 and 64 inclusive. Must be an integer of power of 2. By default, processor groups have a maximum size of 64 logical processors. You can use this boot configuration setting to override the size and makeup of a computer's processor groups for testing purposes. Processor groups provide support for computers with greater than 64 logical processors. This boot option is available on 64-bit versions of Windows 7 and Windows Server 2008 R2 and later versions. This boot option has no effect on the 32-bit versions of Windows 7.

Use the **groupsize** option if you want to force multiple groups and the computer has 64 or fewer active logical processors. For more information about using this option, see Boot Parameters to Test Drivers for Multiple Processor Group Support.

groupaware [**on** | **off**]
Forces drivers to be aware of multiple groups in a multiple processor group environment. Use this option to help expose cross-group incompatibilities in drivers and components. Processor groups provide support for computers with greater than 64 logical processors. This boot option is available on 64-bit versions of Windows 7 and Windows Server 2008 R2 and later versions. This boot option has no effect on the 32-bit versions of Windows 7. You can use the **groupaware** option and the **groupsize** option to test driver compatibility to function with multiple groups when computer has 64 or fewer active logical processors.

The **groupaware on** setting ensures that processes are started in a group other than group 0. This increases the chances of cross-group interaction between drivers and components. The option also modifies the behavior of the legacy functions, **KeSetTargetProcessorDpc**, **KeSetSystemAffinityThreadEx**, and **KeRevertToUserAffinityThreadEx**, so that they always operate on the highest numbered group that contains active logical processors. Drivers that call any of these legacy functions should be changed to call their group-aware counterparts (**KeSetTargetProcessorDpcEx**, **KeSetSystemGroupAffinityThread**, and **KeRevertToUserGroupAffinityThread**).

For more information about using this option, see Boot Parameters to Test Drivers for Multiple Processor Group Support.

maxgroup [**on** | **off**]
Maximizes the number of groups created in a processor group configuration. The **maxgroup on** setting assigns NUMA nodes to groups in a manner that maximizes the number of groups for a particular computer. The number of groups created is either the number of NUMA nodes the computer has, or the maximum number of groups supported by this version of Windows, whichever is smaller. The default behavior (**maxgroup off)** is to pack the NUMA nodes tightly into as few groups as possible.

Use the maxgroup option if you want to use multiple groups, the computer has 64 or fewer active logical processors, and the computer already has multiple NUMA nodes. This option can also be used to alter the default group configuration of a computer that has more than 64 logical processors.

Processor groups provide support for computers with greater than 64 logical processors. This option is available on 64-bit versions of Windows 7 and Windows Server 2008 R2 and later versions. This boot option has no effect on the 32-bit versions of Windows 7.

For more information about using this option, see Boot Parameters to Test Drivers for Multiple Processor Group Support.

onecpu [**on** | **off**]
Forces only the boot CPU to be used in a computer that has more than one logical processor. For example, the following command configures the current operating system loader to use one processor.

syntax	Copy

```
bcdedit /set onecpu on
```

Memory Related Settings

increaseuserva *Megabytes*
Specifies the amount of memory, in megabytes, for user-mode virtual address space.

On 32-bit editions of Windows, applications have 4 gigabyte (GB) of virtual address space available. The virtual address space is divided so that 2 GB is available to the application and the other 2 GB is available only to the system.

The 4-gigabyte tuning feature, enabled with the **increaseuserva** option, allows you to increase the virtual address space that is available to the application up to 3 GB, which reduces the amount available to the system to between 1 and 2 GB. The **BCEdit /set increaseuserva** *Megabytes* command can specify any value between 2048 (2 GB) and 3072 (3 GB) megabytes in decimal notation. Windows uses the remaining address space (4 GB minus the specified amount) as its kernel-mode address space.

See 4-Gigabyte Tuning (Windows) for additional information about this feature.

nolowmem [**on** | **off**] Controls the use of low memory. When **nolowmem on** is specified, this option loads the operating system, device drivers, and all applications into addresses above the 4 GB boundary, and directs Windows to allocate all memory pools at addresses above the 4 GB boundary. Note that the **nolowmem** option is ignored in Windows 8, Windows Server 2012, and later versions of Windows.

pae [**Default** | **ForceEnable** | **ForceDisable**]
Enables or disables Physical Address Extension (PAE). When PAE is enabled, the system loads the PAE version of the Windows kernel.

The **pae** parameter is valid only on boot entries for 32-bit versions of Windows that run on computers with x86-based and x64-based processors. On 32-bit versions of Windows (prior to Windows 8) , PAE is disabled by default. However, Windows automatically enables PAE when the computer is configured for hot-add memory devices in memory ranges beyond the 4 GB region, as defined by the Static Resource Affinity Table (SRAT). *Hot-add memory* supports memory devices that you can add without rebooting or turning off the computer. In this case, because PAE must be enabled when the system starts, it is enabled automatically so that the system can immediately address extended memory that is added between restarts. Hot-add memory is supported only on Windows Server 2008, Datacenter Edition; Windows Server 2008 for Itanium-Based Systems; and on the datacenter and enterprise editions of all later versions of Windows Server. Moreover, for versions of Windows prior to Windows Server 2008, hot-add memory is supported only on computers with an ACPI BIOS, an x86 processor, and specialized hardware. For Windows Server 2008 and later versions of Windows Server, it is supported for all processor architectures.

On a computer that supports hardware-enabled Data Execution Prevention (DEP) and is running a 32-bit version of the Windows operating system that supports DEP, PAE is automatically enabled when DEP is enabled and, on all 32-bit versions of the Windows operating system, PAE is disabled when you disable DEP. To enable PAE when DEP is disabled, you must enable PAE explicitly, by using **/set nx AlwaysOff** and **/set pae ForceEnable**. For more information about DEP, see Boot Parameters to Configure DEP and PAE.

For more information about using the **pae** parameter and the other parameters that affect PAE configuration, see Boot Parameters to Configure DEP and PAE.

removememory *Megabytes*
Removes memory from the total available memory that the operating system can use.

For example, the following command removes 256 MB of memory from the total available to the operating system associated with the specified boot entry.

```
syntax                                                          Copy

bcdedit /set {49916baf-0e08-11db-9af4-000bdbd316a0} removememory 256
```

truncatememory *address* Limits the amount of physical memory available to Windows. When you use this option, Windows ignores all memory at or above the specified physical address. Specify the *address* in bytes.

For example, the following command sets the physical address limit at 1 GB. You can specify the address in decimal (1073741824) or hexadecimal (0x40000000).

```
syntax                                                          Copy

bcdedit /set {49916baf-0e08-11db-9af4-000bdbd316a0} truncatememory 0x40000000
```

VESA, PCI, VGA, and TPM

usefirmwarepcisettings [**yes** | **no**]
Enables or disables the use of BIOS-configured peripheral component interconnect (PCI) resources.

msi [**Default** | **ForceDisable**]
Can be Default or ForceDisable.

vga [**on** | **off**]
Forces the use of the VGA display driver.

novga [**on** | **off**]
Disables the use of VGA modes entirely.

tpmbootentropy [**default** | **ForceEnable** | **ForceDisable**]
Determines whether entropy is gathered from the trusted platform module (TPM) to help seed the random number generator in the operating system.

Processors and APICs

clustermodeaddressing [**integer**]
Defines the maximum number of processors to include in a single Advanced Programmable Interrupt Controller (APIC) cluster.

configflags [**integer**]
Specifies processor-specific configuration flags.

maxproc [**yes** | **no**]
Reports the maximum number of processors in the system.

numproc [**integer**]
Uses only the specified number of processors.

onecpu [**yes** | **no**]
Forces only the boot CPU to be used.

restrictapiccluster [**integer**]
Defines the largest APIC cluster number to be used by the system.

usephysicaldestination [**yes** | **no**]
Forces the use of the physical APIC.

uselegacyapicmode [**yes** | **no**]
Forces legacy APIC mode, even if the processors and chipset support extended APIC mode.

x2apicpolicy [**enable** | **disable** | **default**]
Enables or disables the use of extended APIC mode, if supported. The system defaults to using extended APIC mode if it is available. Can be Enabled, Disabled or Default.

Additional Settings

disabledynamictick [**yes** | **no**]
Enables and disables dynamic timer tick feature.

> ⓘ **Note**
>
> This option should only be used for debugging.

pciexpress [**default** | **forcedisable**]
Enables or disables PCI Express functionality. If the computer platform supports the PCI Express features and the ACPI _OSC method grants control of the features to the operating system, Windows enables the advanced features through the PCI Express Native Control feature (this is the default). Use the **forcedisable** option to override the advanced PCI Express features and use legacy PCI Express behavior. For more information, see [Enabling PCI Express Native Control in Windows](#).

useplatformtick [**yes** | **no**]
Forces the clock to be backed by a platform source, no synthetic timers are allowed. The option is available starting in Windows 8 and Windows Server 2012.

> ⓘ **Note**
>
> This option should only be used for debugging.

xsavedisable [**0** | **1**]
When set to a value other than zero (0), disables XSAVE processor functionality in the kernel.

Debugger Settings

To work with the debugger settings, use the following commands.

⌗ Expand table

Command	Description
BCDEdit /bootdebug	The /bootdebug boot option enables or disables boot debugging of the current or specified Windows operating system boot entry.
BCDEdit /dbgsettings	The /dbgsettings option sets or displays the current global debugger settings for the computer. To enable or disable the kernel debugger, use the BCDEdit /debug option.
BCDEdit /debug	The /debug boot option enables or disables kernel debugging of the Windows operating system associated with the specified boot entry or the current boot entry.

Hypervisor Debugger Settings

Use the **BCDEdit / hypervisorsettings** option to set or display the hypervisor debugger settings for the system. For more information, see [BCDEdit /hypervisorsettings](#).

hypervisordebug [**On** | **Off**]
Controls whether the hypervisor debugger is enabled.

hypervisordebugtype [**SERIAL** | **1394** | **NET**] Can be SERIAL, 1394, or NET. For more information, see [BCDEdit /hypervisorsettings](#).

Hypervisor Settings

hypervisorlaunchtype [**Off** | **Auto**]
Controls the hypervisor launch options. If you are setting up a debugger to debug Hyper-V on a target computer, set this option to **Auto** on the target computer. For more information, see [Create a Virtual Machine with Hyper-V](#).

hypervisorloadoptions NOFORCESNOOP [**Yes** | **No**]
Specifies whether the hypervisor should enforce snoop control on system IOMMUs.

hypervisornumproc *number*
Specifies the total number of logical processors that can be started in the hypervisor.

hypervisorrootproc *number*
Specifies the maximum number of virtual processors in the root partition and limits the number of post-split Non-Uniform Memory Architecture (NUMA) nodes which can have logical processors started in the hypervisor.

hypervisorrootprocpernode *number*
Specifies the total number of virtual processors in the root partition that can be started within a pre-split Non-Uniform Memory Architecture (NUMA) node.

hypervisoruselargevtlb [**yes** | **no**]
Increases virtual Translation Lookaside Buffer (TLB) size.

hypervisoriommupolicy [**default** | **enable** | **disable**]
Controls whether the hypervisor uses an Input Output Memory Management Unit (IOMMU).

Drivers and System Root

driverloadfailurepolicy [**Fatal** | **UseErrorControl**]
Can be Fatal or UseErrorControl.

osdevice [**device**]
Defines the device that contains the system root.

systemroot [**string**]
Defines the path to the system root.

ems [**On** | **Off**]
Enables kernel Emergency Management Services. The BCDEdit /ems option enables or disables kernel Emergency Management Services (EMS) for the specified operating system boot entry. For more information, see BCDEdit /ems.

The BCDEdit /emssettings option sets the global Emergency Management Services (EMS) settings for the computer. For more information, see For more information, see BCDEdit /emssettings.

Virtual Secure Mode

vsmlaunchtype [**Off** | **Auto**]
Controls the Virtual Secure Mode launch type. Can be Off or Auto. For more information, see Manage Windows Defender Credential Guard.

Event Logging

The BCDEdit /event command enables or disables the remote event logging for the specified boot entry. For more information, see BCDEdit /event.

Comments

For more information about specific BCD elements and boot options, you can use the commands **BCDEdit /? OSLOADER** and **BCDEdit /? TYPES OSLOADER**.

To view the current boot entries and their settings, use the **bcdedit /enum** command. This command displays the active boot entries and their associated globally unique identifiers (GUID). Use the identifiers with the **/set** command to configure options for a specific boot entry.

To delete a boot option value that you have set, use the **/deletevalue** option. The syntax for the command is as follows:

bcdedit /deletevalue [{**ID**}] *datatype*

For example, if you change the processor group option, **groupsize**, to a new value for testing purposes, you can revert to the default value of 64 by typing the following command and then restarting the computer.

syntax	Copy

```
bcdedit /deletevalue groupsize
```

Any change to a boot option requires a restart to take effect. For information about commonly used BCDEdit commands, see Boot Configuration Data Editor Frequently Asked Questions.

DTrace

DTrace (DTrace.exe) is a command-line tool that displays system information and events. There is a bcedit option to enable dtrace. For information about the DTrace BCDEdit options available, see the installing section of DTrace on Windows.

Requirements

Minimum supported client: Windows Vista

Minimum supported server: Windows Server 2008

Related topics

- BCDEdit Options Reference
- BCDEdit /deletevalue

Feedback

Was this page helpful? 👍 Yes | 👎 No

Provide product feedback ⧉ | Get help at Microsoft Q&A