

BCD Boot Options Reference

Microsoft Corporation

Boot entry parameters, or boot parameters, are optional, system-specific settings that represent configuration options. You can add boot parameters to a boot entry for an operating system. This describes the boot options for Windows Vista and later versions of Windows that are related to developing, testing, and debugging drivers on computers with x86-based, x64-based, and Itanium-based processors. You can add these parameters to the boot entries for Windows operating systems.

Mapping Boot.ini Options to BCDEdit Options and Elements

The following table provides a mapping from the boot options used in operating systems prior to Windows Vista (in Boot.ini), to the BCDEdit options and the BCD elements used in Windows Vista and later versions of Windows.

Boot.ini	BCDEdit Option	BCD Element Type
/3GB	increaseuserva	BcdOSLoaderInteger_IncreaseUserVa
/BASEVIDEO	vga	BcdOSLoaderBoolean_UseVgaDriver
/BOOTLOG	bootlog	BcdOSLoaderBoolean_BootLogInitialization
/BREAK	halbreakpoint	BcdOSLoaderBoolean_DebuggerHalBreakpoint
/CRASHDEBUG	/dbgsettings /start	
/DEBUG, BOOTDEBUG	/debug /bootdebug	BcdLibraryBoolean_DebuggerEnabled
/DEBUG	/debug	BcdOSLoaderBoolean_KernelDebuggerEnabled
/DEBUG, /DEBUGPORT=	/dbgsettings	BcdLibraryInteger_DebuggerType
/DEBUGPORT=	/dbgsettings <i>DebugType</i> [debugport: <i>Port</i>] [baudrate: <i>Baud</i>] [channel: <i>Channel</i>] [targetname: <i>TargetName</i>] [/start {Active Autoenable Disable} /noemux]	BcdLibraryInteger_SerialDebuggerPort BcdLibraryInteger_SerialDebuggerBaudRate BcdLibraryInteger_1394DebuggerChannel BcdLibraryString_UsbDebuggerTargetName
/EXECUTE	nx	BcdOSLoaderInteger_NxPolicy
/FASTDETECT		
/HAL=	hal	BcdOSLoaderString_HalPath
/KERNEL=	kernel	BcdOSLoaderString_KernelPath
/MAXMEM=	truncatememory	BcdLibraryInteger_TruncatePhysicalMemory
/NODEBUG	/debug	
/NOEXECUTE	nx {	BcdOSLoaderInteger_NxPolicy
/NOGUIBOOT	quietboot	BcdOSLoaderBoolean_DisableBootDisplay
/NOLOWMEM	nolowmem	BcdOSLoaderBoolean_NoLowMemory
/NOPAE	paе	BcdOSLoaderInteger_PAEPolicy
/ONECPU	onecpu	BcdOSLoaderBoolean_UseBootProcessorOnly
/PAE	paе	BcdOSLoaderInteger_PAEPolicy
/PCILOCK	usefirmwarepcisettings	BcdOSLoaderInteger_UseFirmwarePciSettings
/REDIRECT	/ems /emssettings [BIOS] [EMSPORT:{port} EMSBAUDRATE: {baudrate}]]	BcdOSLoaderBoolean_EmsEnabled
/SOS	sos	

BCDEdit /bootdebug

The **/bootdebug** boot option enables or disables boot debugging of the current or specified Windows operating system boot entry.

BCD Boot Options Reference

Microsoft Corporation

```
bcdedit /bootdebug [{ID}] { on | off }
```

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 operating system that is currently active. If a boot entry is specified, the GUID associated with the boot entry must be enclosed in braces **{ }**.

on

Enables boot debugging of the specified boot entry. If a boot entry is not specified, boot debugging is enabled for the current operating system.

off

Disables boot debugging of the specified boot entry. If a boot entry is not specified, boot debugging is disabled for the current operating system.

Comments

The **/bootdebug** boot option enables boot debugging for a specific boot entry. Use the **/dbgsettings** option to configure the type of debugging connection (*debugtype*) to use and the connection parameters. If no **/dbgsettings** are specified for the boot entry, the global debug settings are used. The default values for the global settings are shown in the following table.

dbgsetting parameter	Default value
Debugtype	Serial
Debugport	1
Baudrate	115200

The following command disables boot debugging of Windows Boot Manager (*Bootmgr.exe*). Windows Boot Manager selects which operating system will start, and then loads the Windows boot loader.

```
bcdedit /bootdebug {bootmgr} off
```

The following command enables boot debugging of the Windows boot loader for the current operating system. The Windows boot loader (*Winload.exe*) controls the progress bar and loads the kernel boot drivers.

```
bcdedit /bootdebug on
```

In the following example, the first command sets the global debugger settings for a 1394 kernel debugging connection. The next three commands enable debugging of Windows Boot Manager, the boot loader, and then kernel debugging of the operating system. This combination allows debugging at every stage of startup. If this combination is used, the target computer will break into the debugger three times: when Windows Boot Manager loads, when the boot loader loads, and when the operating system starts up.

BCD Boot Options Reference

Microsoft Corporation

```
bcdedit /dbgsettings 1394 CHANNEL:1
bcdedit /bootdebug {bootmgr} on
bcdedit /bootdebug on
bcdedit /debug on
```

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 /dbgsettings SERIAL [DEBUGPORT:port] [BAUDRATE:baud] [/start
startpolicy] [/noumex]
```

```
bcdedit /dbgsettings 1394 [CHANNEL:channel] [/start startpolicy] [/noumex]
```

```
bcdedit /dbgsettings USB TARGETNAME:targetname [/start startpolicy]
[/noumex]
```

Parameters

SERIAL

Specifies that the target machine and the host machine will use a serial connection for debugging. When this option is used, the **DEBUGPORT** and **BAUDRATE** parameters can be included as well.

1394

Specifies that the target machine and the host machine will use an IEEE 1394 (FireWire) connection for debugging. When this option is used, the **CHANNEL** parameter can be included as well.

USB

Specifies that the target machine and the host machine will use a USB 2.0 connection for debugging. When this option is used, the **TARGETNAME** parameter must be included as well.

DEBUGPORT:port

(Only used when the connection type is **SERIAL**.) Specifies the serial port to use as the debugging port. This is an optional setting. The default port is **1** (COM 1).

BAUDRATE:baud

(Only used when the connection type is **SERIAL**.) Specifies the baud rate to use. This parameter is optional. Valid values for *baud* are 9600, 19200, 38400, 57600, and 115200. The default baud rate is 115200 bps.

Note If the Windows Special Administration Console (SAC) application is running on a target machine that is configured for kernel mode debug through a serial port, the SAC application may cause the debugger connection to drop. This event occurs because the COM port baud value changes after the debugger connection is established. Either close the SAC application before running the debugger or change the debugger COM port baud value to 9600.

BCD Boot Options Reference

Microsoft Corporation

CHANNEL:*channel*

(Only used when the connection type is **1394**.) Specifies the 1394 channel to use. The value for *channel* must be a decimal integer between 0 and 62, inclusive, and must match the channel number used by the host computer. The channel specified in this parameter does not depend on the physical 1394 port chosen on the adapter. The default value for *channel* is 0.

TARGETNAME:*targetname*

(Only used when the connection type is **USB**.) Specifies a string value to use for the target name. This string can be any value.

/start *startpolicy*

This option specifies the debugger start policy. The following table shows the options for the *startpolicy*.

<i>startpolicy</i>	Description
ACTIVE	Specifies that the kernel debugger is active.
AUTOENABLE	Specifies that the kernel debugger is enabled automatically when an exception or other critical event occurs. Until then, the debugger is active but is disabled.
DISABLE.	Specifies that the kernel debugger is enabled when you type kdbgctrl to clear the enable block. Until then, the debugger is active but is disabled.

If a start policy is not specified, ACTIVE is the default.

/noumex

Specifies that the kernel debugger ignores user-mode exceptions. By default, the kernel debugger breaks for certain user-mode exceptions, such as STATUS_BREAKPOINT and STATUS_SINGLE_STEP. The **/noumex** parameter is effective only when there is no user-mode debugger attached to the process.

Comments

The **/dbgsettings** option configures the global debugging settings, but does not enable debugging. You must use the **/debug** option to enable debugging for a specific boot entry. If there are no debugging settings specified for a particular boot entry, the global debug settings are used. To override the global settings, you must use the **BCDEdit /set** command and specify the ID of the boot entry along with the debug parameter and value pair.

The default values for the global settings are serial communications using COM1, at a baud rate of 115,200.

/dbgsetting parameter	Default value
Connection type	Serial
DEBUGPORT: <i>port</i>	1
BAUDRATE: <i>rate</i>	115200

BCD Boot Options Reference

Microsoft Corporation

Examples

The following example configures the global debugger settings to use serial communications using COM2 and a baud rate of 115,200.

```
bcdedit /dbgsettings serial debugport:2 baudrate:115200
```

In the following example, the first command sets the global debugger settings for USB 2.0 and names the target myVistaTarget. The second command enables the debugger for the current session.

```
bcdedit /dbgsettings usb targetname:myVistaTarget  
bcdedit /debug ON
```

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.

```
bcdedit /debug [{ID}] { on | off }
```

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 operating system that is currently active. If a boot entry is specified, the GUID associated with the boot entry must be enclosed in braces **{ }**.

on

Enables kernel debugging of the specified boot entry. If a boot entry is not specified, kernel debugging is enabled for the current operating system.

off

Disables kernel debugger of the specified boot entry. If a boot entry is not specified, kernel debugging is disabled for the current operating system.

Comments

The **/debug** boot option enables kernel debugging for a specific boot entry. Use the **/dbgsettings** option to configure the type of debugging connection to use and the connection parameters. If no **/dbgsettings** are specified for the boot entry, the global debug settings are used. The default values for the global settings are shown in the following table.

dbgsetting parameter	Default value
Connection type	Serial
Debug port	1
Baud rate	115200

The following example enables kernel debugging of the boot entry with the GUID of 49916baf-0e08-11db-9af4-000bdbc316a0.

```
bcdedit /debug {49916baf-0e08-11db-9af4-000bdbc316a0} on
```

BCD Boot Options Reference

Microsoft Corporation

In the following example, the first command sets the global debugger settings for USB 2.0 and names the target myVistaTarget. The second command enables the debugger for the current session.

```
bcdedit /dbgsettings usb targetname:myVistaTarget
bcdedit /debug ON
```

BCDEdit /ems

The **/ems** option enables or disables Emergency Management Services (EMS) for the specified operating system boot entry.

```
bcdedit /ems [{ID}] { on | off }
```

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 **{ }**.

Comments

In Windows Vista and later, use **BCDEdit /emssettings** command and its parameters to establish EMS settings for all boot entries. Then, use the **BCDEdit /ems** command to enable EMS for a particular boot entry.

EMS allows users to control particular components of a server remotely, even when the server is not connected to the network or to other standard remote-administration tools. For information about EMS, search for Emergency Management Services on the Microsoft TechNet Web site.

Example

The following command enables EMS for a boot entry with the identifier of {49916baf-0e08-11db-9af4-000bdbc316a0}.

```
bcdedit /ems {49916baf-0e08-11db-9af4-000bdbc316a0} on
```

BCDEdit /emssettings

The **/emssettings** option sets the global Emergency Management Services (EMS) settings for the computer. To enable or disable EMS, use the **/ems** option. The **/emssettings** option does not enable or disable EMS for any boot entry.

```
bcdedit /emssettings [ BIOS ] | [ EMSPORT: port | [EMSBAUDRATE: baudrate] ]
```

Parameters

BIOS

Specifies that the system will use BIOS settings for the EMS configuration. This works only on systems that have EMS support provided by the BIOS.

EMSPORT: *port*

BCD Boot Options Reference

Microsoft Corporation

Specifies the serial port to use as the EMS port. This parameter should not be specified with the **BIOS** option.

EMSBAUDRATE: *baudrate*

Specifies the serial baud rate to use for EMS. This command should not be specified with the BIOS. The *baudrate* is optional, and the default is 9,600 bps.

Comments

To properly enable EMS console redirection after Windows is installed, Windows needs to know the port and transmission rate that the computer uses for out-of-band communication. Windows uses these same settings for EMS console redirection.

On computers with BIOS firmware and an ACPI Serial Port Console Redirection (SPCR) table, Windows can find the out-of-band settings established in the BIOS by reading entries in the SPCR table. On these systems, you can use the **BIOS** parameter to direct Windows to look in the SPCR table for the port settings, or you can use the **emsport:port** and **emsbaudrate:baudrate** parameters to override the settings in the SPCR table.

On computers that have BIOS firmware, but do not have an SPCR table, use BCDEdit and the **/emssettings** command with the **emsport:port** parameter to specify the port and with the **emsbaudrate:baudrate** parameter to specify the transmission rate.

On all systems, use the **BCDEdit /ems** command and specify the boot entry to enable EMS console redirection on the operating system that the boot entry loads.

The boot parameters described in this section enable EMS console redirection after Windows is installed. For information about enabling EMS during a new installation or upgrade of Windows, search for "Enabling Emergency Management Services" on the Microsoft TechNet Web site. For a detailed example, see Boot Parameters to Enable EMS Redirection.

BCDEdit /set

The **/set** command sets a boot entry option value in the boot configuration data store. Use this command to configure specific boot entry elements, such as kernel debugger settings, data execution protection (DEP) and processor address extension (PAE) options, and to load alternate hardware abstraction layer (HAL) and kernel files.

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.

Possible values of *datatype value*

bootlog [**yes** | **no**]

BCD Boot Options Reference

Microsoft Corporation

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.

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. 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*.

hal *file*

Directs the operating system loader to load an alternate HAL file. The specified file must be located in the *%SystemRoot%\system32* directory, and its file name must conform to 8.3-character format.

increaseuserva *Megabytes*

Specifies the amount of memory, in megabytes, for user-mode virtual address space. This variable can have 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.

BCD Boot Options Reference

Microsoft Corporation

kernel file

Directs the operating system loader to load an alternate kernel. The specified file must be located in the %SystemRoot%\system32 directory, and its file name must conform to the `kernel` format.

loadoptions busparams=*Bus.Device.Function*

Specifies the target controller when multiple controllers exist. This syntax is appropriate when using either a 1394 cable or a USB 2.0 debug cable for debugging. *Bus* specifies the bus number, *Device* specifies the device number, and *Function* specifies the function number.

Note For 1394 debugging, the bus parameters must be specified in decimal, regardless of which version of Windows is being configured. The format of the bus parameters used for USB 2.0 debugging depends on the Windows version. In Windows Vista and Windows Server 2008, the USB 2.0 bus parameters must be specified in hexadecimal. In Windows 7 and later versions of Windows, the USB 2.0 bus parameters must be specified in decimal.

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 this 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](#).

lowmem [on | off]

Controls the use of low memory. When **lowmem 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.

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 how to use the Control Panel to change the DEP settings, see the [Change Data Execution Prevention settings](#) page on the Windows Help and How-to Web site.

BCD Boot Options Reference

Microsoft Corporation

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.

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.

```
bcdedit /set onecpu on
```

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, 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 2003, Datacenter Edition; Windows Server 2003, Enterprise Edition; Windows Server 2008, Datacenter Edition; Windows Server 2008, Enterprise IA64 Edition; 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.

BCD Boot Options Reference

Microsoft Corporation

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, except Windows Server 2003 with SP1, 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.

PAE is required to support Cache Coherent Non-Uniform Memory Architecture (known as ccNUMA or NUMA) on computers with x86 processors, although Windows can run in non-NUMA mode on NUMA-capable computers without PAE. Even when it is required, PAE is not enabled automatically. NUMA is supported in all editions of Windows XP and Windows Vista, and in the Enterprise and Datacenter editions of Windows Server 2003.

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

If you are running a 32-bit version of Windows Vista on a computer that has 4 GB of memory installed, the amount of RAM available for use could be less than what you would expect. For more information about memory usage, see article Q929605, "The system memory that is reported in the System Information dialog box in Windows Vista is less than you expect if 4 GB of RAM is installed" in the Microsoft Knowledge Base.

quietboot [on | off]

Controls the display of a high-resolution bitmap in place of the Windows boot screen display and animation. In operating systems prior to Windows Vista, the **/noguiboot** serves a similar function.

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.

```
bcdedit /set {49916baf-0e08-11db-9af4-000bdbc316a0} removememory 256
```

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.

testsigning [on | off]

Controls whether 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 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.

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.

BCD Boot Options Reference

Microsoft Corporation

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

```
bcdedit /set {49916baf-0e08-11db-9af4-000bdbc316a0} truncatememory 0x40000000
```

usefirmwarepcisettings [yes | no]

Enables or disables the use of BIOS-configured peripheral component interconnect (PCI) resources.

vga [on | off]

Forces the use of the VGA display driver.

Comment

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 [{ID}] /deletevalue datatotype
```

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.

```
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.