

Bcdedit Basics

Mark Minasi

(Reprinted from WindowsItPro Magazine)

Executive Summary

Windows Server 2008 and Windows Vista have changed the way you control Windows' boot loader. With this change comes a new tool, Bcdedit, that every systems administrator must master.

Anyone who uses Windows Server 2008 or Windows Vista knows that these latest Windows versions have brought some significant changes to Microsoft's OSs. One of these changes is the way you control Windows' boot loader, the software that supports storing more than one OS on a drive and that lets you configure the way Windows boots. With this change comes a new tool, Bcdedit, that anyone who administers Server 2008 or Vista systems must master.

Prior to Vista, we controlled boot order and boot options by modifying the boot.ini text file at the root of the C drive. To change the boot-loader behavior of pre-Vista systems, we needed nothing more complex than Notepad and the knowledge of how to un-set boot.ini's "read only" component. Now, Microsoft has replaced boot.ini with a binary file called BCD, which resides in a hidden/system folder called \boot on whatever drive you boot your system from. Bcdedit is the tool you'll use to control BCD.

Boot Alterations

The BCD boot-loader structure introduces some new terminology. Whereas a boot.ini file might have several sections that let you boot a particular OS with specific options, Bcdedit has boot entries. A new Server 2008 or Vista installation has just one boot entry, but if you add an OS—for example, if you install both Server 2008 and Vista on the same system, or if you have only one OS on your system but want the option to boot that OS with different startup or debugging settings—you'd have more than one boot entry. You can see your boot entries by opening an elevated command prompt and typing :

```
bcdedit /v
```

The top section of the output that Figure 1 shows, labeled Windows Boot Manager, contains global information: which boot entry boots by default, how long users have to choose a non-default boot entry, and so on.

Bcdedit Basics

Mark Minasi

(Reprinted from WindowsItPro Magazine)

Windows Boot Manager

```
identifier {9dea862c-5cdd-4e70-acc1-f32b344d4795}
device partition=D:
description Windows Boot Manager
locale en-US
inherit {7ea2e1ac-2e61-4728-aaa3-896d9d0a9f0e}
default {c96b3ca5-f287-11dc-84e2-e41e15845d6e}
resumeobject {c96b3ca6-f287-11dc-84e2-e41e15845d6e}
displayorder {6443870f-3c0c-11dd-835a-c0389998b4ec}
toolsdisplayorder {c96b3ca5-f287-11dc-84e2-e41e15845d6e}
timeout {b2721d73-1db4-4c62-bf78-c548a880142d}
10
```

Windows Boot Loader

```
identifier {6443870f-3c0c-11dd-835a-c0389998b4ec}
device partition=E:
path \Windows\system32\winload.exe
description Microsoft Windows Server 2008
locale en-US
inherit {6efb52bf-1766-41db-a6b3-0ee5eff72bd7}
osdevice partition=E:
systemroot \Windows
resumeobject {64438710-3c0c-11dd-835a-c0389998b4ec}
nx OptOut
hypervisorlaunchtype Auto
```

Windows Boot Loader

```
identifier {c96b3ca5-f287-11dc-84e2-e41e15845d6e}
device partition=C:
path \Windows\system32\winload.exe
description Microsoft Windows Vista
locale en-US
inherit {6efb52bf-1766-41db-a6b3-0ee5eff72bd7}
osdevice partition=C:
systemroot \Windows
resumeobject {c96b3ca6-f287-11dc-84e2-e41e15845d6e}
nx OptIn
```

Below that, Bcdedit displays two boot entries labeled Windows Boot Loader. The first boots a copy of Server 2008 (as you can see in the description line), and the second boots Vista.

Notice that Bcdedit /v shows a lot of long hexadecimal strings enclosed in brackets—items that you probably recognize as globally unique identifiers (GUIDs). You can safely ignore most of these GUIDs, but pay special attention to the first one in each boot entry, labeled identifier. You need these GUIDs when you inform Bcdedit which boot entry to make the default, what order to display the boot entries, which boot entries to copy, and so on. If you're like me, you'll find yourself feeling thankful that you can mark text and put it into the clipboard.

To tell BCD which boot entry to load by default, you use the Bcdedit /default command. For example, to tell my system to boot the Server 2008 boot entry, I would type

```
bcdedit /default {6443870f-3c0c-11dd-835a-c0389998b4ec}
```

Bcdedit Basics

Mark Minasi

(Reprinted from WindowsItPro Magazine)

If you can't imagine typing or even copying those GUIDs, you'll be glad to know that Bcdedit recognizes two well-known GUIDs: The {current} value is the boot entry that the OS is currently running, and the {default} value is whatever boot entry BCD currently knows about.

If I were to delete the \Windows folder from my Server 2008 system's E drive, I would essentially uninstall Server 2008 from my system—but BCD wouldn't know that. So, every time I booted my system, I'd see a useless option to boot Server 2008. To configure BCD to no longer show the Server 2008 option, I could use the Bcdedit /delete command:

```
bcdedit /delete {6443870f-3c0c-11dd-835a-c0389998b4ec}
```

If you're thinking about trying some of these commands, I recommend creating an extra boot entry to play with. Believe me, you don't want to accidentally delete your main OS's boot entry! To make a copy of your boot entry for test purposes, you can use the Bcdedit /copy command:

```
bcdedit /copy {current} /d "Test entry"
```

Note that I've made use of the {current} well-known GUID and added the /d option (mandatory with the /copy switch), which let me give the option a name for the new OS boot entry.

Whenever I set up a Windows NT 3.1 or later system, I find boot.ini's default 30-second wait to be excessive, unacceptably slowing down my boot times. So, I always like to modify the "timeout=30" value to 5 seconds. To do the equivalent in Bcdedit, I type

```
bcdedit /timeout 5
```

Familiarize Yourself

I miss boot.ini's simplicity, but BCD and Bcdedit are here to stay, so I recommend spending a little time getting familiar with Bcdedit. I'll cover more in-depth options next time.

Booting Up with Bcdedit

In "Bcdedit Basics", I introduced you to Bcdedit, a tool in Windows Vista and later that lets you modify the way your system boots. (The tool's name refers to Microsoft's replacement of the boot.ini text file with a binary data file called BCD.) In that article, you learned how to store information in the BCD about how to boot more than one OS, but even if you have only one OS on a computer, you'll still find Bcdedit useful: It also lets you modify what Microsoft calls *OS entry options* (or what I refer to as *boot options*).

Boot.ini used to let you boot pre-Vista systems under some non-standard set of parameters. For example, Windows NT has always offered the /vga boot option, which—if placed in a system's boot.ini file—instructs the OS to boot using not a hardware-specific display driver but rather the generic SVGA driver. (The ability to choose an OS entry is useful if you realize too late that you've installed a defective or mismatched video driver.)

Bcdedit's *OS entry options* work similarly. You can control how your system boots with the Bcdedit /set command, as follows:

```
bcdedit /set <OS entry> <boot option> <boot option value>
```

Bcdedit Basics

Mark Minasi

(Reprinted from WindowsItPro Magazine)

That looks ugly, so how about a concrete example? The following Bcdedit /set command would instruct Bcdedit that your copy of Vista should always use the generic SVGA driver rather than a board-specific driver:

```
bcdedit /set {9c219fb1-bb55-11dd-97ac-804080387aa6} vga yes
```

As you can see, I first identified the OS entry to add. Recall from last month that Bcdedit identifies different OS options not with user-defined words or phrases but with ugly GUIDs randomly generated by the OS. (You can see the GUIDs associated with each OS entry by using the Bcdedit /v command.) Remember, these GUIDs are random numbers; don't just type in the GUID I've entered above, as it's almost certainly not the correct GUID for your system.

After the OS entry GUID, I've entered a pair of values: the name of a parameter that controls how the system boots, followed by the value of that parameter; *vga=yes* would be a clearer way to type it, but that's not how Microsoft designed Bcdedit's syntax.

Now, in the real world, you probably wouldn't want your system to always use the SVGA driver. So how might you make use of that Bcdedit command? How about using the Bcdedit /copy command to copy the OS entry that you normally use to boot your system to a second entry called something like Vista Safer Mode? It's not Safe Mode, per se, but it's the same basic idea, and you can define exactly what Vista Safer Mode means. Then, once you've created the new Vista Safer Mode OS entry, you can use its GUID to construct the Bcdedit /set command that adds **vga=yes** to that OS entry. Now, just type *bcdedit* to see your OS entries. You'll see one that looks something like Figure 1.

Once you see that output, it's easy to see that OS entries already have a number of boot options, such as **device** and **osdevice** (which help tell the OS option what drive to boot from and where to find the Windows loader), and *description*.

But what if you decide that you want to remove the *vga* boot option? Of course, you could just change its value from yes to no, but not every boot option offers that sort of flexibility. In that case, use the Bcdedit /deletevalue command:

```
bcdedit <OS entry> <boot option name>
```

To remove the *vga=yes* option altogether, you'd type

```
bcdedit /deletevalue {9c219fb1-bb55-11dd-97ac-804080387aa6} vga
```

Now that you've seen one boot option (e.g., *vga*), the logical next question is, "What other options can I use?" There are quite a few, and we'll examine those next month.

Bcdedit Options

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 Basics

Mark Minasi

(Reprinted from WindowsItPro Magazine)

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 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

Bcdedit Basics

Mark Minasi

(Reprinted from WindowsItPro Magazine)

	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 BCDEdit, 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 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.

Bcdedit Basics

Mark Minasi

(Reprinted from WindowsItPro Magazine)

/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
```

-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.