

28 April 2007

By: Marius Oiaga, Technology News Editor

Windows Vista
Microsoft

[Advanced Boot Architecture Management in Windows Vista](#)

For dual-boot and multi-boot configurations

1. Introduction to the New Windows Vista Boot Architecture Advanced boot architecture management in Windows Vista has to begin with a basic exploration and understanding of the boot environment introduced by the operating system. Windows Vista brings to the table a pre-operating system boot architecture. First off, this means that the old and familiar Boot.ini file is not longer a part of the boot environment, and when it is not ignored it is completely missing. And along with Boot.ini Windows Vista has also lost Bootcfg used for boot configuration editing. This is true and not quite. Let me explain. Bootcfg is still a valid command prompt in Windows Vista. Simply open a new command prompt window with administrative privileges (in order to do so type “cmd” in the Search Box integrated in the Start Menu and then, with the “cmd” returned result selected press Ctrl + Shift + Enter). Enter “Bootcfg /?” in the command prompt window and you will be able, even in Windows Vista, to access a description of the tool along with all the parameters that can be used. However, users will be informed from the start that the command line tool can be used exclusively to configure, query, change or delete the boot entry settings in the Boot.ini file. As Windows Vista no longer uses the boot.ini file, the command line tool that will be used with this version of the Windows platform is BCDEDIT. The overhauled boot loader architecture in Windows Vista features the Boot Configuration Data (BCD) as a core component. BCD is a boot system designed to deliver firmware-independent configuration and storage, adding both security and increased performance to the boot process. And if Bootcfg is obsolete in Vista you will have to turn to the also new boot configuration command editing tool BCDEDIT.

2. New Windows Vista Boot Loader Components The Windows NT boot loader is simply just another component that has not survived into Windows Vista. Instead, the operating system includes the following boot loader trinity: Windows Boot Manager (Bootmgr.exe), Windows operating system loader (Winload.exe) and Windows resume loader (Winresume.exe). Windows Vista no longer differentiates between BIOS-based and EFI-based computers. Older versions of Windows were centered around either the MBR disk partitioning style and BIOS for 32-bit platforms or the GUID Partition Table disk partitioning style and Extensible Firmware Interface for 64-bit. The Windows boot process is controlled by Ntldr and Boot.ini for x86 and by Ia64ldr.efi, Diskpart.efi and Nvrboot.efi for x64. On Windows Vista, and this is also valid for Windows Server Longhorn, Boot Configuration Data (BCD) delivers storage for boot options for both BIOS-based and EFI-based computers, that previously stored boot configuration parameters in a text file and respectively in the NVRAM. In this context, BCD brings together the best out of the two boot scenarios on Windows Vista. Microsoft's design of the BCD involved creating a common, firmware-independent and platform-independent boot option interface for the new operating system.

3. The Tools of the Windows Vista Boot “Trade” Now that we have established that Boot Configuration Data is the heart of the Windows Vista boot loader architecture, let's take a look at the system tools that enable users to access and manage BCD. I have already mentioned BCDEDIT as the replacement of Bootcfg. In Windows Vista, BCDEDIT.EXE is the ultimate command-line tool to manage the boot configuration data store. I am now paraphrasing the description provided by Windows Vista when “bcdedit /?” is entered into a command prompt windows that runs with elevated privileges. The Windows Vista Boot Configuration Store Editor can add, delete, edit, and append entries in the BCD store. According to Microsoft, the BCD 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).” But I have already dealt with this aspect, haven't I? BCDEDIT.EXE is in a sense synonymous with advance boot configuration data store management. In fact, Windows Vista features two additional tools that are a tad more approachable and intuitive: Startup and Recovery and the System Configuration utility.

3.a The One, the Only... BCDEDIT.EXE While Windows Vista still includes Bootcfg.exe for support of older operating systems, BCDEDIT.EXE is the most advanced command-line utility for boot configuration data store management in the operating system. Access is simple, just launch a command prompt windows with administrator privileges and you are all set. Administrator privileges are a must as otherwise the UAC will deny you access to the tool. Since bcdedit is a command-line utility, all your work will be centralized in cmd. Here is an insight into the basic functionality of bcdedit, courtesy of Microsoft. Bcdedit.exe currently enables you to do the following:

- Create a BCD store for a later installation of Windows Server "Longhorn".
- Add entries to an existing BCD store
- Modify existing entries in a BCD store.
- Delete entries from a BCD store.
- Export entries to a BCD store.
- Import entries from a BCD store.
- List currently active settings.
- Query entries of a particular type.
- Apply a global change (to all the entries).
- Change the default time-out value.

In order to familiarize yourselves with all that Bcdedit.exe has to offer, there is a simple command that you can enter in the command prompt window, and that is “bcdedit /?”. As I have mentioned above, this command will permit you to access a description of the tool but also the complete list of commands that can be used to manage the Windows Vista boot configuration data store.

3.b. Startup and Recovery Startup and Recovery can be accessed via Start, Control Panel, System and Maintenance, System and “Advanced system settings” on the left hand side of the Tasks menu. You will be promoted for administrator privileges, so make sure you hit OK. System Properties, the Advanced tab displays a Startup and Recovery area with “System startup, system failure and debugging information” and a Settings button. Pressing this button will take you to the Startup and Recovery dialog box, just as in the image on the left. From here, users can select the default operating system for their machine and control the amount of time the boot process will display the list with the available operating systems. Additionally, users can set Vista's behavior in the eventuality of a system failure, including compiling an event system log and the actions to be taken, implicitly a restart. Debugging information settings can also be modified from the Startup and Recovery dialog box. Here is Microsoft's description of the tool “the Startup and recovery dialog box enables you to select the default operating system to start if you have multiple operating systems installed on your computer. You can also change the time-out value (for listing available operating systems).”

3.c. Msconfig.exe – the System Configuration Utility Msconfig.exe raises the bar a little on the way to the advanced BCDEDIT.EXE. The utility can be executed by entering “msconfig” in the Search Box of the Start Menu, hitting the Enter key and then the OK button on the User Account Control prompt. System configuration is simply at the first glance more complex than the Startup and Recovery dialog box. First off, users will be able to control the Startup method by checking the box next to the options available including Normal startup, Diagnostic startup and Selective Startup. System Configuration also allows users to enable or disable services, configure the Startup Items and access the built in tools of Windows Vista. Via the Boot tab, you will be able to manage the current operating systems installed on your computer, set the default operating system, manage boot options and timeout.

4. Windows Vista Boot Configuration – Down, Dirty and Advanced Windows Vista boot configuration data store can be expansively managed via BCDEDIT.EXE, but do so responsibly and make sure that you understand the results of every action and command as you risk throwing the operating system in a non-bootable state. It is a healthy precaution measure to backup Vista before you proceed. Just in case...

4.a. Set the Default Operating System In a scenario where you have a dual-boot or a multi-boot configuration on your computer you need to be in control of what operating system will load by default, once the timeout runs out. But in order to do so, you must first find the ID of the operating system you want to set

as default. This can be done using the “bcdedit /enum all” command. The identifiers are placed in {} and Microsoft delivers this example “{466f5a88-0af2-4f76-9038-095b170dc21c}.” In essence, the identifier does nothing more than to point to the GUID. After you have identified the ID of the operating system that you wish to set as default, run this command “bcdedit /default ID”. Microsoft’s own example is bcdedit /default {466f5a88-0af2-4f76-9038-095b170dc21c}. But how to identify which operating system is which in a multi-boot configuration? When you run “bcdedit /enum all” it will provide information related to the Windows Boot Manager and Windows Boot Loader and so on and so forth, but the Windows Boot Loader will contain a description entry that will say Microsoft Windows Vista, as of course you have booted into Vista or otherwise you wouldn’t be using bcdedit. The Windows Boot Manager will feature several Ids depending on how many operating systems you have installed on your computer under displayorder. This displayorder is nothing else than the display order of the operating systems that you are presented when you have to select the OS to boot in. The operating systems that are older than Windows Vista will be featured under Windows legacy OS Loader.

4.b. Increase or Decrease the Boot Manager Time-OutIn Windows Vista, all the operating systems that are available for boot will be displayed for of a total of 30 seconds before the default operating system will boot. This number can be altered from both the Startup and Recovery dialog box and from the msconfig, but also from bcdedit. In this context, in order to change the time that a computer will wait for a decision from the user before the default operating system is selected, users have to enter the following command “bcdedit /timeout TimeOut” where TimeOut is a numeric value. For example, enter “bcdedit /timeout 10” if you want the computer to wait for just 10 seconds before booting the default OS instead of 30.

4.c. Alter the Boot Sequence for the Next RebootKeeping in mind what I said under 4.a. about ID, the command prompt for a custom boot sequence for the next time that you restart the machine is “bcdedit /bootsequence {ID} {ID} {ID}...” This command is helpful if you have booted into one of the operating systems installed on your machine and then when you restart you want to automatically select another operating system. Keep in mind that this command is only valid for the next restart. Otherwise, the restart order is governed by DISPLAYORDER.

4.d. Mix It Up in the DisplayorderThe /displayorder parameter governs the list of the operating systems installed on the local machine. You will find that this list is illustrated under displayorder entry in Windows Boot Manager. As all the other parts of the Windows Vista boot configuration data store, displayorder is also opened to modifications. If for any reason you are not happy with the way your machine displays the installed operating systems, then you can simply set the boot manager display order according to your preferences. Unlike altering the boot sequence for just the next restart with the /bootsequence parameter this change is permanent, or until you decide to modify it again. For this you can use the following command “bcdedit /display {ID} {ID1} {ID2}...” where ID is a value that specifies the GUIDs that will end up in the display order of the operating systems. Additionally, you can use “bcdedit /displayorder {ID} [/addlast/addfirst/remove]” to introduce or remove an operating system in the boot manager display order. Here is an example provided by the Redmond Company: “bcdedit.exe /displayorder {c84b751a-ff09-11d9-9e6e-0030482375e6}-addlast”.