



Boot Technology in Windows Vista and Windows 7

By [Jack Chen](#), Sr. Windows Consultant and Practice Leader

The average computer user takes for granted the fact that everytime a computer is started or rebooted, it must load the operating system. The process is relatively transparent to the user, but the details of this process vary depending on the system's hardware and firmware and whether the system is booted from a disk drive, a USB drive, a network location (PXE), or some other source.

Microsoft has entirely redesigned the boot technology for Windows Vista and later versions (including Windows 7) to deal with the growing complexity and variety of new hardware and firmware. This new technology in the boot process is managed by the BCD store (the boot configuration data store) which contains the boot configuration parameters and controls the computer's boot environment. This new BCD fully affects the boot process in Windows Vista and Windows 7, compared to the Boot.ini which influences the boot process in Windows XP and Windows 2000.

We have learned that the BCD gives you full control of what will happen during the Windows Vista or Windows 7 boot process. Without understanding the BCD in Windows Vista (NT6 based) and Windows 7 (NT7 based), a Windows administrator might not be able to customize the operating system boot parameters when needed (i.e. Dual boot, Safe boot, Crash debug) and might not even be able to troubleshoot and/or optimize parameters for a server's boot configuration options (i.e. MaxProcsPerCluster, Pcilock, Userva=).

When you start your computer with either Windows Vista (Windows 2008) or Windows 7 (Windows 2008 R2) installed, the BIOS is first executed to initialize various devices which include processors, memory, disks, and all other I/O. The BIOS will then read the MBR (master boot record) from the boot device (which is usually the first sector on the disk) and transfer control over the bootstrap code stored in the MBR. Then the boot manager (bootmgr) reads the boot menu from the BCD (Boot Configuration Data) store, so it can reach the loader and display a boot menu to the end users. After the Boot Manager (bootmgr) displays the boot menu, if there is more than one operating system installed, end users can select either operating system to load before the specified boot menu times out. Afterward, the Boot Manager (bootmgr) transfers control over the Boot Loader (Winload) to load the OS kernel and device drivers for the execution environment.

In summary, the new boot applications in Windows Vista and Windows 7 include:

- Bootmgr: An API that controls boot flow. In a multiboot system it displays an OS boot menu.
- Winload: An OS loader. It creates the execution environment for the operating system and also loads the kernel and device drivers.
- Winresume.exe: An OS resume loader. It restores Windows to its running state from hibernation.

The new BCD WMI provider for Windows 7 and Windows 2008 R2 add support for qualified partitions and virtual hard disks. A qualified partition is a physical partition that is uniquely identified by its disk signature. A virtual hard disk (VHD) can be used in the same ways as a physical hard disk, including as a boot device. If you need more information about this particular new BCD WMI provider, please read About VHD in Microsoft TechNet.

Now, if we look back at Windows XP or Windows 2000 and compare them with the new operating systems Windows Vista and Windows 7, it's easy to see that they are dramatically different.

For a PC/AT BIOS computer, which is being booted from its hard drive and running Windows XP or Windows 2000, the BIOS first reads the MBR and then transfers control over the boot code - ntldr which later loads Windows. The

TAOS

boot.ini is the component that is used to control how ntldr loads the OS. The boot.ini is located on the root folder of the boot device. It contains a separate boot entry for each OS version (known as ARC path, such as multi0,disk0,rdisk0,partition1) if it is dual boot or more, and it also contains Windows boot options (such as BASEVIDEO, FASTDETECT) that are available for end users. End users just need to select their option from the list in the boot menu to load OS.

All those parameters were previously in the Boot.ini file, but now you can use Bcdedit.exe to add, delete and modify entries in the BCD store. For detailed command and option information, type bcdedit.exe /? <command>. For example, to display detailed information, type: bcdedit.exe /ENUM; to display detailed information about the /createstore command, type: bcdedit.exe /? /createstore. You can always run the following command to list entries in the BCD store:

```
C:\>bcdedit /enum /v
```

Windows Boot Manager

```
identifier      {9dea862c-5cdd-4e70-acc1-f32b344d4795}
device          partition=C:
description     Windows Boot Manager
locale          en-US
inherit         {7ea2e1ac-2e61-4728-aaa3-896d9d0a9f0e}
default         {f9d7f773-6149-11de-bab4-cede933c175b}
resumeobject    {f9d7f773-6149-11de-bab4-cede933c175b}
displayorder    {f9d7f774-6149-11de-bab4-cede933c175b}
toolsdisplayorder {b2721d73-1db4-4c62-bf78-c548a880142d}
timeout         90
```

Windows Boot Loader

```
identifier      {f9d7f774-6149-11de-bab4-cede933c175b}
device          partition=C:
path            \Windows\system32\winload.exe
description     Windows Server 2008 R2
locale          en-US
inherit         {6efb52bf-1766-41db-a6b3-0ee5eff72bd7}
```



```
recoverysequence    {f9d7f775-6149-11de-bab4-cede933c175b}
recoveryenabled     Yes
osdevice            partition=C:
systemroot          \Windows
resumeobject        {f9d7f773-6149-11de-bab4-cede933c175b}
nx                  OptOut
hypervisorlaunchtype Auto
```

The following information digested from Microsoft TechNet provides an overview and describes how to use the related tools to manage boot options. As a reminder, the BCDEdit is a command-line tool that is used to add, delete, and modify data in the BCD store. You must use your GUID and test by running bcdedit.exe.

Specify the Default Operating System

To specify the default operating system, use:

```
bcdedit /default ID
```

ID is the GUID for the Windows boot loader boot entry that is associated with the desired operating system. For example:

```
bcdedit /default {cbd971bf-b7b8-4885-951a-fa03044f5d71}
```

To change the default boot entry to the legacy loader, set ID to {ntldr}, which is BCDEdit's well-known name for the GUID that is associated with Ntldr.

```
bcdedit /default {ntldr}
```

The following Microsoft Visual Basic Script sample shows how to use the BCD WMI API to specify the default operating system. It takes a single argument, the GUID that is associated with the boot entry for the new default operating system.

```
set Locator = CreateObject("WbemScripting.SWbemLocator")
set Services = Locator.ConnectServer(".", "root\wmi")
Services.Security_.ImpersonationLevel = 3
```

```
DefaultOsIdentifier = WScript.Arguments(0)
```

```
'These hardcoded values will be replaced with official constants
' when available.
```

```
const BootMgrId = "{f9d7f775-6149-11de-bab4-cede933c175b}"
```

```
const DefaultType = &h23000003
```



```
' Open up a connection to WMI BcdStore class, allowing for  
' impersonation. We need to request that Backup and Restore  
' privileges be granted as well.
```

```
set BcdStoreClass = GetObject("winmgmts:{impersonationlevel=impersonate,(Backup,Restore)}!" & MachineName &  
"root/wmi:BcdStore")
```

```
if not BcdStoreClass.OpenStore("", BcdStore) then  
    WScript.Echo "Couldn't open the system store!"  
    WScript.Quit  
end if
```

```
' Open the "boot manager" object.
```

```
if not BcdStore.OpenObject(BootMgrId, BootMgr) then  
    WScript.Echo "Couldn't open the boot manager object!"  
    WScript.Quit  
end if
```

```
' Set the boot manager's default OS object to the specified OS.  
' Note that objects must be passed as strings.
```

```
if not BootMgr.SetObjectElement(DefaultType, DefaultOSIdentifier) then  
    WScript.Echo "Couldn't set the default OS value!"  
    WScript.Quit  
end if
```

```
WScript.Echo "Successfully set the boot manager's default OS value."
```

Specify the Boot Manager's Timeout Value

To specify the boot manager's timeout value, use:

```
bcdedit /timeout Timeout
```

Timeout specifies the value in seconds. For example, to specify a 15-second timeout value:

```
bcdedit /timeout 15
```

Change a Boot Entry's Description

The description is the text that appears in the list of boot entries that is displayed to the user at boot time. Use the following command to change a boot entry's description. *ID* is the GUID that is associated with the desired boot entry.

```
Bcdedit /set ID description "The new description"
```

For example:

```
bcdedit /set {f9d7f775-6149-11de-bab4-cede933c175b} description "My Favorite OS"
```

Create a New Windows Vista Boot Entry

The following procedure creates an additional Windows Vista boot entry. This allows a user, for example, to have separate normal and debug configurations for the same version of the operating system.

1. Make a copy of an existing Windows Vista boot entry, as shown in the following example. *ID* is the GUID that is associated with the boot entry to be copied. BCDEdit creates a GUID for the new boot entry.

```
Bcdedit /copy ID /d "New entry description"
```

2. The preceding command returns the GUID that is associated with the new boot entry. Use that GUID to modify the partition information, as shown in the following example. *NewID* is the GUID of the new boot entry, and this example sets the partition to "d:".

```
Bcdedit /set NewID device partition=d:
```

```
Bcdedit /set NewID osdevice partition=d:
```

3. Add the new boot entry to the display order. The following example adds it to the end of the list.

```
Bcdedit /displayorder NewID –addlast
```

4. Make any additional configuration changes that are required, such as enabling the kernel debugger.

Delete a Boot Entry

The following command deletes a boot entry from BCD. *ID* is the GUID that is associated with the boot entry.

```
bcdedit /delete ID
```

Control How Boot Entries Appear to the User

To specify the order in which boot entries appear to the user, run the following command. *ID1*, *ID2*, and so on are the GUIDs that are associated with the boot entries. Any boot entries that are not included in the list do not appear. If only one entry is specified, the Windows boot manager simply selects that entry without displaying a list.

```
bcdedit /displayorder ID1 [ID2] [ID3] [...]
```

The following command specifies three boot entries: two identified by their GUIDS and Ntldr by its well-known name.

```
bcdedit /displayorder {f9d7f775-6149-11de-bab4-cede933c175b}{f9d7f775-6149-11de-bab4-cede933c175b}  
{cbd971bf-b7b8-4885-951a-fa03044f5d71} {ntldr}
```

The following command adds a boot entry to the beginning or end of the current list, or removes an entry from the list.

```
bcdedit /displayorder ID [/addlast] [/addfirst] [/remove]
```

The following example adds an Ntldr entry to the end of the display order.

```
bcdedit /displayorder {ntldr} /addlast
```

It is also possible to specify a display order that applies only to the next reboot. After that, BCD reverts to the original display order. Use the following command, where the IDs are the GUIDs that are associated with the boot entries.

```
bcdedit /bootsequence ID1 [ID2] [ID3] ...
```

Enable Kernel Debugging

Use the following BCDEdit command to enable or disable kernel debugging for a specified boot entry.

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

ID is the GUID that is associated with a boot entry. If it is omitted, BCDEdit modifies the current boot entry by default. To specify a particular boot entry, set ID to the string form of the associated GUID.

The following example enables kernel debugging for the specified entry.

```
bcdedit /debug {cbd971bf-b7b8-4885-951a-fa03044f5d71} on
```

How to modify BCD when installing a Windows XP onto a computer running Windows Vista

To install a Windows XP operating system on a computer running Windows Vista, use the following procedure.

1. Install the previous version of Windows.
2. Log on to the older operating system and restore the latest boot manager by running the following. Fixntfs.exe will be in the \boot directory of the active partition.

```
fixntfs /lh
```

3. Create a BCD entry for the older operating system by specifying the following. Bcdedit.exe is located in the \Windows\System32 directory of the Windows Vista partition. Description is the description of the new entry for the older operating system.

```
Bcdedit /create {legacy} /d "Description"
```

```
Bcdedit /set {legacy} device boot
```

```
Bcdedit /set {legacy} path \ntldr
```

```
Bcdedit /displayorder {legacy} /addlast
```

4. Restart the computer in order for the changes to take effect.

Resources

[Introduction to Boot Process](#)