

Correction(s)

This procedure remains valid and may be used as originally published. However, two people have told me it was not necessary for them. Both had new Toshiba laptops with Vista pre-installed and added SUSE 10.2 to that environment. At this time I do not know of other exceptions though I presume others do exist.

The procedure has been modified to support a normal install if it will work. Step 7 of the procedure now includes a test to see whether the rest of the procedure is unnecessary.

I have some information which may explain why a normal install works for some people but not enough to predict it. Please see the Microsoft document available at download.microsoft.com/download/9/c/5/9c5b2167-8017-4bae-9fde-d599bac8184a/OEMBoot_Vista.doc and the references provided therein. Also, I would love to hear from others who had no problem installing Linux on top of a Production release of Vista to make a dual boot environment without using the technique described herein or one of the other methods discussed.

Please email details to [lthielster at gmail dot com](mailto:lthielster@gmail.com) (figure it out). Again, you should only do so if you loaded linux by normal methods on top of a live production version of Vista. Please include hardware models, Vista version (including if OEM installed). Linux distro and release. If possible paste the output of `bdcdedit /enum` from Vista as well as a hex dump of your MBR[s] into the email. Other information such as a partition list (`sfdisk -l` in Linux) and the content of your `/boot/grub/menu.lst` would be welcome as well.

The Issues

Historically it has been easy to dual boot PCs with different OS's (Operating Systems) because of well understood standards for how the boot process works. Most modern Linux distributions could be installed along side Windows and could automatically resize any Windows partitions, replace the MBR (Master Boot Record) and adapt a menu to permit the user to boot either Windows or Linux.

Unfortunately Microsoft has chosen to break the standard with Vista¹. There is no point in debating the reasons an 800 pound gorilla did something; one simply finds ways to live with it. The hard part is separating the good information on the web from the information outdated by the production release of Microsoft Windows Vista. A large percentage of the information on the net is wrong simply because it was developed during beta test and RC (Release Candidate) phases. Microsoft made important changes for the production release versions of Vista which invalidated the older information. The clearest and most succinct description of those particular changes I have found is:

¹ There are other issues as well, but this is the real deal breaker. Some of the other issues will come up later in the article.

extract from

http://blogs.sun.com/moinakg/entry/vista_and_solaris_express_dual

Windows Vista (final release version) has the following new quirks

1. Vista installer requires that no partition be active (or bootable) on the harddisk! If an active partition is found the Vista installer simply says something like "This Volume is not suitable for installing Windows" and refuses to proceed!!
2. Vista requires that the Windows partition remain active in order to boot.
3. Vista requires that the Vista bootloader be present in the MBR in order to boot. If you put any other bootloader in the MBR (like Grub stage1), Vista will refuse to boot saying that winload.exe is corrupt. This is because Vista's loader checks for a disk signature in the MBR! More info is available here:
<http://www.windowsbbs.com/showthread.php?t=55415>. Microsoft talks about it in a whitepaper giving some weird reasons:
http://www.google.co.in/search?q=OEMBoot_Vista.
4. The boot.ini file has been replaced by adding yet another entry in the Registry called the Boot Configuration Data with a CLI utility called bcdedit.exe to allow fiddling with the BCD.

To my knowledge the Beta and RC1 versions of Vista did not have restriction **#3**, but it is present in the final release candidate version.

Clearly, these conditions make the traditional methods for dual booting unusable. There does seem to be an exception which is quite interesting. Apparently, one can use two disk drives² with the Vista drive being the second and when the user chooses to boot Vista the boot process remaps the drives so Vista thinks it is on the first drive. For more information, see:

http://wiki.gtwy.net/index.php/Dual_Boot_Vista_and_Linux

Additionally, many of the resources on the web require the user to install Vista to a partition substantially smaller than the full hard drive. This is not an option for users who get Vista pre-installed on a new computer. Generally those users receive only a restore or recovery disk rather than a full Vista Install disk. They have no ability to [re-]install Vista other than as the only OS on the hard disk and

² Most laptop users would not be able to use the dual drive technique.

using the entire disk³. Another issue for them is that Vista is not friendly to some of the traditional tools for resizing NTFS partitions such as that on which Vista resides. Fortunately, Vista itself resolves this problem by being able to shrink its own partition while running (more later).

Overview

The method described in this How-to was mostly adapted from the web page referenced in the inset on page 1. The adaptations were largely to make it usable by people with OEM installed Vista, generalize it away from the Solaris specifics, and to add a convenience partition useful to some dual-booters. It is not specific to OEM installed Vista machines, it should be usable by anyone. As written, it assumes Vista is loaded before Linux but it should be easy to adapt to the reverse (except for OEM limited systems).

Generically, the method is to:

- Ensure the system can be restored,
- Shrink the Vista partition leaving unallocated space for Linux,
- Save a copy of the Vista MBR,
- Install Linux,
- Save a copy of the Linux MBR (we assume for GRUB),
- Restore the Vista MBR and make that partition bootable,
- Reboot to Vista,
- Make the GRUB MBR available to Vista,
- Add registry entries so the Vista Boot Loader also knows how to invoke GRUB, and
- Enjoy a well deserved cool one!

Downsides include leaving the Vista boot loader in place and the complexity of doing the install this way. You will need some sort of Linux LiveCD which includes the dd utility (make sure before you begin) and a USB flash drive or some other storage media usable by both Vista and Linux.

How to do it

Some steps are labeled as optional. While not necessary to this procedure, you may find them useful and in a logical position for this procedure.

1. Log in to Vista as a user with administrative privileges such as the initial [or only] user.
2. Do a complete backup unless this is a freshly installed Vista. If it was pre-installed by the OEM, be sure you have the restore/recovery DVD/CDs or make them before proceeding.

³ The exception being those OEMs who provide a Recovery partition on the hard disk and use the rest for Vista.

3. Set a Restore point.
4. Resize the Vista partition.
 - a) Go to Disk Management (click the Start dot thingie, type "com" in the search box, select Computer Management from the Menu and the Disk Management in the resulting window).
 - b) Right click the Vista drive (normally C:) in the upper or lower central pane and select the "Shrink Volume" option. Once Vista figures out how much it is able to shrink the volume, you must choose an amount such that Vista is left with enough free space for your needs **and** there is enough space freed for Linux and your needs there. Note you may wish to also leave space for the optional partition in the next step.
 - c) Wait for Vista to do the resize which should leave unallocated space on the drive.
5. OPTION: Make a separate partition which can be read and written by both OSs and possibly even made into network shared storage.
 - a) Right click the Unallocated Space and select the "Create Partition" entry. FAT32 is a good format for sharing and the size is entirely dependent on your intended usage (such as a shared music store or file transfer medium) and the amount of available space. Be sure to leave enough unallocated space for Linux. Vista will give it a drive letter such as D: which, while meaningless to Linux, may be convenient to remember for naming a mount point in Linux.

Later, in Linux, you can access this partition by mounting it. Assuming the partition were known as /dev/hda2 in Linux and D: to Vista, one might use "**sudo mkdir /mnt/D**" to make a directory in which to mount the partition and "**sudo mount /dev/hda2 /mnt/D**" to actually mount it. Note the name of D is a convenience for its similarity to the Vista name but could be anything you like. Also, this method is for a temporary mount. How to make a permanent mount is beyond the scope of this article.
6. Save a copy of the Vista MBR⁴.
 - a) Reboot the machine using a Linux LiveCD (such as Knoppix, Ubuntu, or a SystemRescue CD) with the dd utility on it. If your machine does not boot from the CD, you may have to press a key at startup or even make a change to BIOS to boot from a CD. See your system documentation for more information.
 - b) Open a terminal window and verify the device name for the drive containing Vista. It is most likely /dev/hda or /dev/sda. Enter "**sudo**

⁴ The mbr_xxxx.bin files are important! You might save copies to more than one device (e.g. flash drive AND hard disc).

fdisk -l" and look for the partition identified as having an HPFS/NTFS file system. The drive is the same device name as the partition but without the numeric digit on the end.

NOTE: If you created the optional partition in step 4, you might wish to note its referent. For example, it might be /dev/hda2 and it could be mounted for use as noted above.

- c) Assuming the drive was /dev/hda, enter the command **sudo dd if=/dev/hda of=~/Desktop/mbr_vista.bin bs=512 count=1** being very careful to double check the command before pressing the ENTER key. This should place the file mbr_vista.bin on your desktop.
 - d) Move the file to persistent storage. Since you are running a LiveCD Linux, the desktop does not represent persistent storage. You could insert a USB flash drive and drag the file to it. If you created the FAT32 partition in the OPTIONAL step above you could mount that and drag the file there. In any case, this must be a storage location available later when the LiveCD OS is no longer running.
7. Install your preferred Linux distribution to the unallocated space on the drive and, if given a choice, let GRUB be installed to the MBR. NOTE: this may make Vista unbootable but we fix that below.
- a) No instructions for installing your favored Linux are given since each is different.
 - b) Test the result. For the present, the machine may boot only to Linux. Try rebooting to Linux and Vista. Even if the grub menu shows Vista or Longhorn as an option, selecting it may not work as Vista may detect that it's proprietary MBR is not in place and refuse to boot. If you are able to boot to either you are done. If you can only boot to Linux, continue with this procedure.
 - c) OPTION: if you created the optional partition in step 4, now would be a handy time to mount it (especially if you did so in step 5 and stored mbr_vista.bin there).
8. Save a copy of GRUB's MBR.
- a) In a terminal window, and again assuming /dev/hda was the boot device, enter the command:
sudo dd if=/dev/hda of=~/Desktop/mbr_grub.bin bs=512 count=1
Recheck your spelling before hitting enter. This should create the file mbr_grub.bin on your Linux Desktop.
 - b) Although this is persistent storage, we need to eventually put it somewhere Vista can see it. For now, copy it to that USB flash drive or the OPTIONAL FAT32 partition you may have created above.
9. Restore most of Vista's MBR and make Vista bootable again. Restoring the

entire MBR as Vista had it would eliminate the partitioning done to support Linux. But Vista **requires** its own MBR be present so we will restore the first 446 bytes and leave the partition table alone.

a) Mount the persistent storage on which you placed mbr_vista.bin in step 5 above, open a terminal and cd to that location. A simple **"ls mbr*"** should show the file to be present.

b) Issue the command

sudo dd if=mbr_vista.bin of=/dev/hda bs=446 count=1

and check your typing and that you specified 446 before hitting the ENTER key.

c) Use parted to make the Vista partition bootable again. Assuming the Vista partition is on /dev/hda, do the following in a terminal:

sudo parted /dev/hda (run the partition editor on device /dev/hda)

print (print partition table to see which Vista is on)

set 1 boot on (set the boot flag for that partition)

print (verify it was done)

quit

Following this operation, Vista is once again bootable, but Linux is not.

d) OPTION: Re-save the Vista MBR since it now has the correct partition table data. This is not used for this procedure, but it seems a good idea. Use the command

sudo dd if=/dev/hda of=~/Desktop/mbr_vista_new.bin bs=512 count=1

to use a new name for the file.

10.Reboot the machine. It should boot to Vista with no boot menu (except perhaps its own).

11.Set another Restore point.

12.Copy mbr_grub.bin to somewhere accessible to Vista. [C:\mbr_grub.bin](#) is fine but it could be in any folder or drive that Vista can read.

13.Add GRUB to the Vista boot menu. Note that Vista uses registry entries for boot configuration data rather than boot.ini as did previous versions of Windows. Vista provides a special command line program called bcdedit for editing these special registry entries. You will need to run it as administrator.

a) Click on the START button replacement, type "com" into the search box, right click on the "Command Prompt" entry and select "Run as administrator". You will need to grant permission to proceed.

b) Create a registry entry for grub in the vista boot menu. Doing so will return one of those long ids you often see in the windows registry. To create the registry entry and get the id, enter the command

bcdedit /create /d "GRUB" /application BOOTSECTOR

though you can substitute any description for the "GRUB" shown here as it is simply the line that will show up in the menu. You should see the {grubID} value printed on the screen. It will look something like {dee6b505-c5ac-11db-a7c6-001534e41bb9} but we will refer here to it as {grubID}. You need to accurately record it as you will use it repeatedly. It would probably also be helpful to have it visible in another window from which you can copy it and paste it into the command window (use a right click to do so). Wherever you see {grubID} in the following steps, you use the ID you recorded. Be sure to save it.

- c) Add which drive contains the grub MBR with the command
bcdedit /set {grubID} device partition=C: (or whatever drive⁵ is correct)
- d) Add the path on that device using
bcdedit /set {grubID} path \mbr_grub.bin (or whatever path is correct⁶)
- e) Tell Vista where in the menu list to show the new entry with
bcdedit /set displayorder {grubID} /addlast
- f) Set the timeout value for the menu in seconds
bcdedit /timeout 10
- g) OPTION: You should be able to set the default menu choice to be grub using
bcdedit {bootmgr} /set default {grubID}
but I have only tried this once. It did work for me.

That should be it. At this point any bootup of the machine should display the Windows boot menu and selecting GRUB from that menu should display a normal grub menu. If you select Longhorn or Vista from that menu, it will take you back to the Windows boot menu.

⁵ Isn't it interesting how Vista seems to be mixing Windows and Linux terminology here?

⁶ I do not know what limitations Vista has for the path so keep it simple.