

# EFI-Booting Ubuntu on a Mac

by Rod Smith, [rodsmith@rodsbooks.com](mailto:rodsmith@rodsbooks.com)

Originally written: 1/2011; Last Web page update: 6/17/2013 (last major update: 5/4/2012)

I'm a technical writer and consultant specializing in Linux technologies. This Web page is provided free of charge and with no annoying outside ads; however, I did take time to prepare it, and Web hosting does cost money. If you find this Web page useful, please consider making a small donation to help keep this site up and running. Thanks!

Donate \$1.00	Donate \$2.50	Donate \$5.00	Donate \$10.00	Donate another value
<input type="button" value="Donate"/>	<input type="button" value="Donate"/>	<input type="button" value="Donate"/>	<input type="button" value="Donate"/>	<input type="button" value="Donate"/>

**Note:** This page is written using a rather elderly 32-bit Mac Mini as a reference, and using Ubuntu 12.04 as a reference. Developments in the last year have rendered certain of the procedures on this page sub-optimal. I've tried to point these out, but I haven't fully researched better replacements, and I lack the modern hardware on which to test some of the better methods on more recent 64-bit Macs. Thus, you may need to deviate from these instructions on modern computers.

## The Problem

When installing Ubuntu Linux on an Intel-based Macintosh, most people follow any of several guides available on the Internet, such as [this Ubuntu document](#). Most of these guides, however, rely on features of the Macintosh that are intended to enable it to boot Windows. In particular, these guides typically end up creating a configuration that boots Linux using its *Compatibility Support Module (CSM)*, which provides BIOS emulation on the Mac. In this mode, the Mac sets up a Basic Input Output System (BIOS) layer that the OS can use, rather than rely on the [Extensible Firmware Interface \(EFI\)](#) that OS X uses to run on the computer. BIOS emulation is expedient, and it's the only way that most versions of Windows can boot directly on a Mac, but it's got several drawbacks:

- **Longer boot times**—BIOS emulation is widely regarded as slowing the boot process. I've not tested this effect myself, so I'm not sure how significant it is.
- **Reduced graphics card flexibility**—On some models with multiple graphics chipsets, you can't select which chipset to use in BIOS mode, but you can in EFI mode. My own Mac isn't one of the affected models, so I can't comment further on this issue.
- **Use of a hybrid MBR**—Intel-based Macs use the [GUID Partition Table \(GPT\)](#) rather than the more common [Master Boot Record \(MBR\)](#) partitioning system. Windows, however, can't boot from GPT disks on BIOS-based computers (which it thinks a Mac is thanks to Apple's BIOS emulation). Therefore, Apple uses an ugly and dangerous hack known as a [hybrid MBR](#), in which a GPT data structure known as the *protective MBR* is altered to make the disk look like an MBR disk to Windows. The trouble is that hybrid MBRs frequently cause problems. In fact, even the Ubuntu installer often gets it wrong; it tries to be helpful by creating a hybrid MBR, but depending on your partition layout, it can create a partition table that the popular libparted-based partitioning tools won't touch.

As far as I can tell, BIOS emulation mode only works when a hybrid MBR is present on the hard disk or when a BIOS-bootable optical disk is inserted in the optical drive. Thus, you can't use BIOS emulation mode with a conventional GPT disk unless you plan to use a bootable optical disc to boot Linux.

Ubuntu can use the BIOS emulation and hybrid MBR that enable Windows to boot on Macs, *but it doesn't need to!* If you don't use these features, you won't see their problems. Hybrid MBRs, in particular, are a Bad Idea (with a capital B and a capital I). I've seen numerous problems reported at the [Ubuntu Forums](#) related to Ubuntu installations on Macs with hybrid MBRs, and more problems with similar configurations elsewhere. See my [hybrid MBR Web page](#) for a technical description of what they are and what can go wrong with them. In short, they can get out of sync, cause confusion about which partition is which, and become damaged in frustrating ways. Whenever I think I've heard of every possible problem with hybrid MBRs, I stumble across a new one. You're better off without one, if that's possible—and it is, if you dual-boot OS X and Linux but *not* Windows.

The trouble is that Ubuntu doesn't support installing to a Mac in EFI mode, at least not as of version 12.04.

(Some sources claim that this now works on some later versions, but I haven't tested this claim—indeed, I lack the hardware to do so!) The normal 64-bit discs can boot on PCs in either BIOS mode or (if the PC supports it) UEFI mode (UEFI being, essentially, EFI 2.x). These discs reportedly do not boot on Macs (although I suspect they might with the help of [rEFIt](#) or [rEFInd](#)). The 32-bit Ubuntu installation discs lack EFI support. Ubuntu has also released 64-bit disc images for Macs. Ironically, these boot only in BIOS mode! Therefore, to rid your computer of the dangerous hybrid MBR, you'll need to jump through some hoops. An alternative to all of this is to run Ubuntu in a virtual environment, such as [VirtualBox](#). This can be a good solution in some cases, but it doesn't give Linux direct access to the hardware and you'll lose some speed.

**Note:** My own Mac Mini is an early 32-bit model. Therefore, I can't speak to issues that are specific to 64-bit Macs. I believe the instructions on this page will work for them, although you should pay attention to the caveats presented in the next section.

## Some Caveats

Although my opinion is that Intel-based Macs are best booted using EFI and a GPT with a protective MBR rather than a hybrid MBR, there are drawbacks to doing it this way. These include:

- **No Windows**—As noted earlier, Windows relies heavily on BIOS emulation to boot on a Mac; therefore, if you want to triple-boot OS X, Linux, and Windows, you'll need a hybrid MBR to boot Windows. (You might, however, consider dual-booting OS X and Linux and run Windows in a virtual machine such as [VirtualBox](#). This will eliminate the need for BIOS emulation and hybrid MBRs.) In theory, you could triple-boot with a hybrid MBR but still use EFI mode to boot Linux. I've not tested this, and the benefits are greatly reduced since you've still got the hybrid MBR on your hard disk.
- **Limited nVidia driver support**—The biggest caveat to booting Linux in EFI mode on Macs is that some Linux video drivers, particularly for nVidia chipsets, don't work well when you boot in EFI mode. Thus, if you've got a Mac with an nVidia video chipset, you'll need to use the less capable fbdev drivers rather than the faster nVidia drivers. My own first-generation Intel Mac Mini uses an Intel video chipset and is not affected by this problem, so it doesn't bother me, but it could be a deal-breaker for you. I've seen some suggestions that these problems are less serious now (in April of 2012) than they were a year or two ago. The [UEFIbooting page](#) has some notes on specific Mac models' video support when booted in EFI mode; see the table under "Tested Configurations" about 2/3 of the way down the page.
- **Missing hardware features**—I've seen reports that Linux features such as screen brightness control and suspend/resume may not work correctly when booting using EFI. I don't use such features, so I can't comment personally.
- **Virtual terminals**—I've seen reports that text-mode virtual terminals don't work from an EFI boot; however, this limitation doesn't apply to my computer. Perhaps it interacts with the video chipset, framebuffer driver use, kernel version, or some other variable.

**Note:** Enthusiasts have been working to get Windows to boot in EFI mode on Macs, with some progress and considerable pain. See [this thread on MacRumors](#), for example. This goal appears to be more attainable with Windows 8 than with Windows 7, and success depends on your specific Mac model.

- **OS architecture limitations**—Older Intel-based Macs, such as my Mac Mini, use 32-bit CPUs and EFIs. Newer models come with 64-bit CPUs and EFIs. If you try to run a 32-bit distribution on a 64-bit EFI, you may not have access to EFI Runtime Services. This can reportedly limit some features, but I'm a bit unclear about the details. In my opinion, it's best to stick with 64-bit distributions on 64-bit hardware even on BIOS-based computers, so I don't see this as a big problem.

If you're running Fedora, CentOS, Mandriva, OpenSUSE, Gentoo, Debian, or most other Linux distributions, the procedure I'm about to describe will probably work; however, some details may differ. In fact, at one time or another, I've installed Ubuntu, CentOS, and OpenSUSE on my Mac Mini, all using EFI-mode boots. It's possible that your preferred distribution already supports direct EFI installation, in which case the problem of installing to a Mac in EFI mode may not exist. Check your distribution's documentation to be sure.

I've tested this procedure only on my first-generation Intel-based Mac Mini. This computer uses a 32-bit CPU and 32-bit EFI, so I've tested only the 32-bit version of Ubuntu. It's possible that the 64-bit version has different requirements, or you may need to do things differently on newer 64-bit Macs. You can check which type of EFI you've got with the following command, typed in an OS X Terminal:

```
$ ioreg -l -p IODeviceTree | grep firmware-abi
```

The result should include the string `EFI32` or `EFI64`; the meaning of each should be obvious.

## Prerequisites

Before proceeding, you'll need to download a few items:

- **Ubuntu Linux**—You can obtain Ubuntu from [its main Web site](#). The Web site describes the 32-bit version as "recommended," but lacks such a description for the 64-bit version. Don't worry, though; the 64-bit version works fine for almost everything. Remember that you'll probably need the Mac-specific version to boot on a Mac if you've got a 64-bit system. This version can be harder to track down; I used [this torrent list](#). I recommend using the 64-bit version if you've got a 64-bit EFI, as described earlier. (Note that the string `AMD64` appears in many places in reference to the 64-bit version of Ubuntu, but this version works fine on Intel's x86-64 CPUs; the `AMD64` string is used simply because AMD invented the architecture.) I used a desktop version of Ubuntu 12.04 as a reference for this Web page, but the basic procedures should be similar for the server or alternate installer, as well as for Ubuntu variants such as Kubuntu and Edubuntu. I strongly recommend installing Ubuntu 12.04.2 or a more recent version rather than the original 12.04, since more recent versions come with kernels that include an EFI stub loader, which can simplify certain configuration procedures. Burn the disc and check that it includes directories called `boot`, `install`, `isolinux`, and so on. If you prefer to use a USB flash drive or similar device, you can do so. See Ubuntu's [Installing Ubuntu Desktop](#) page for instructions on creating bootable USB flash drives under various OSes.
- **GPT fdisk (gdisk)**—Go to the GPT fdisk [Sourceforge download page](#) and download the Mac (`gdisk-version.pkg`) package. Install this package in OS X. Once you have Linux installed, you may want to install the Linux version from the Ubuntu repositories (it's in a package called `gdisk`.)
- **Super GRUB 2 Disk**—Download this handy CD image from [its home page](#). Be sure to get Super GRUB 2 Disk, not Super GRUB Disk. Burn the disc image like you burned the Ubuntu install disc image, or copy it to a second USB flash drive.
- **rEFInd**—This is a fork of the earlier rEFIt Mac boot manager. It's part of the boot process as described on this Web page. Download either the ISO image or the binary `.zip` file from [its Web page](#). If you're already

**Note:** For some reason, although I was able to install Ubuntu 10.10 from a CD to my 32-bit Mac Mini, the Ubuntu 12.04 CD hung. A USB flash drive made from the same image file booted fine with the help of rEFInd, though. If you have this problem, you'll have to install rEFInd before you can install Ubuntu. The upcoming section, [Fixing the Installation](#), describes rEFInd installation.

using rEFIt, you can continue to use it rather than install rEFInd; however, rEFIt appears to have been abandoned, and rEFInd adds many improvements, so you may want to upgrade sooner or later.

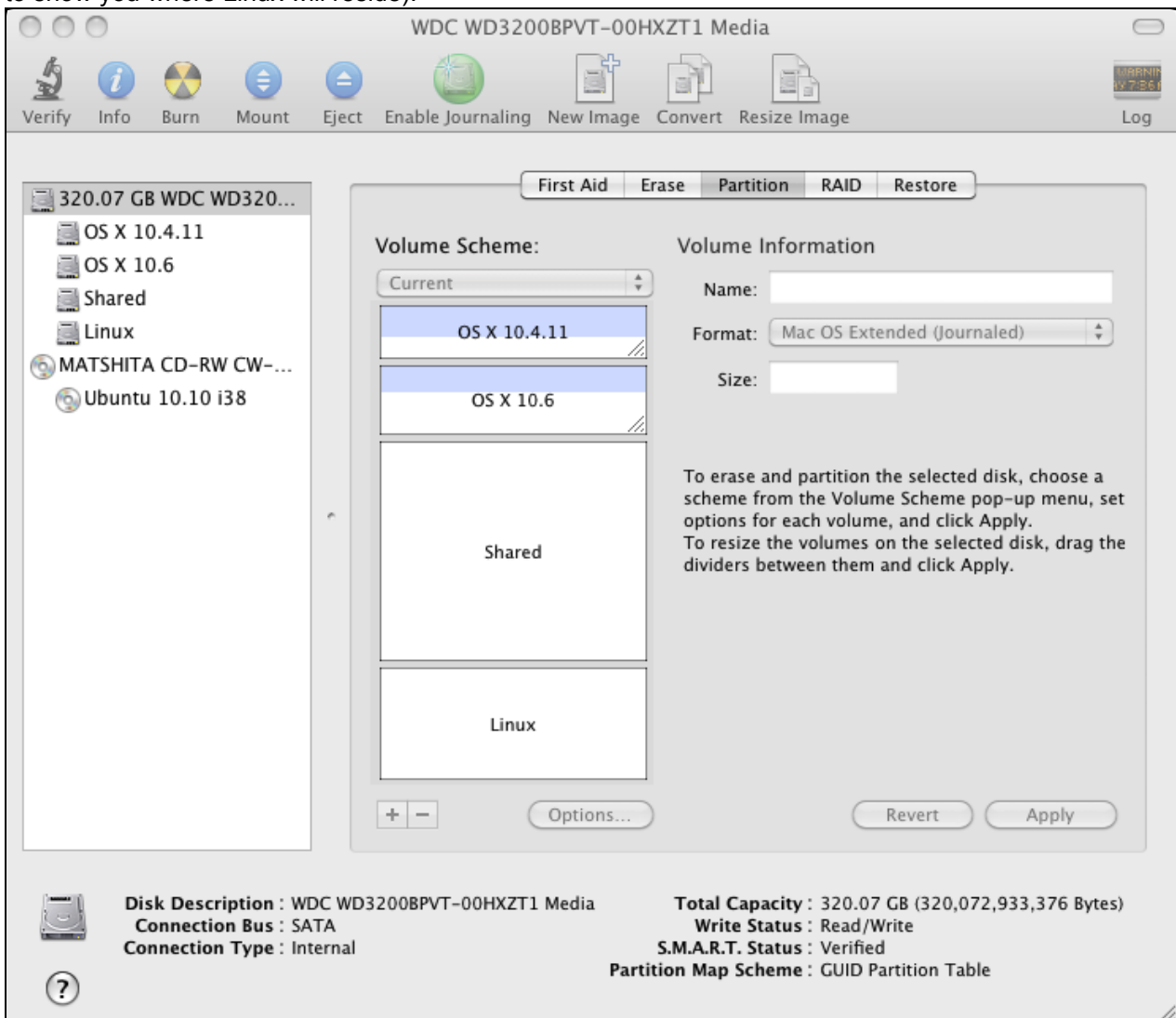
With these items in hand, you can proceed with installing Ubuntu, and then fix it up so that it doesn't use a hybrid MBR.

## Installing Ubuntu

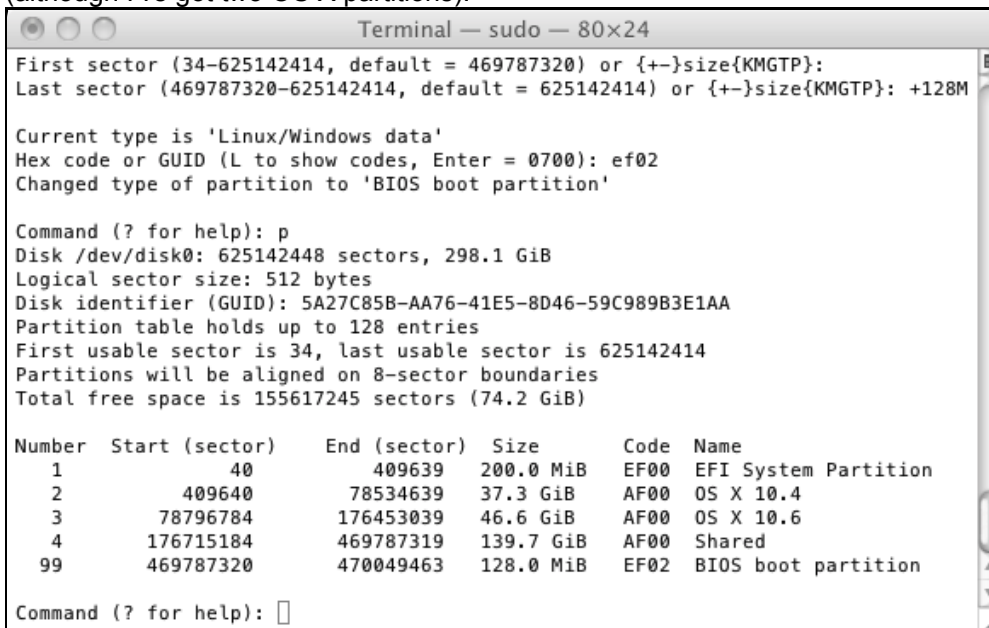
If you've already got a working dual-boot configuration with OS X and Ubuntu, you can skip this section and jump ahead to ["Fixing the Installation."](#) You may be able to skip some of the steps in that section, too, such as rEFInd installation.

To install Ubuntu Linux 12.04 on an Intel-based Mac, follow these directions:

1. If necessary, install Mac OS X.
2. Boot into Mac OS X.
3. If necessary, using OS X's Disk Utility, adjust the sizes of your partitions, making room for Linux. Note that resizing partitions is inherently dangerous, so back up your data before beginning! You may also want to create a FAT or *unjournaled* HFS+ ("Mac OS Extended") shared-data partition. The result might look like the below (the Linux partition in this screen shot could as easily be blank space; I've created it here mainly to show you where Linux will reside).



4. If you haven't done so already, install the OS X version of GPT fdisk (`gdisk`).
5. Open a Terminal program.
6. In the Terminal program, type `sudo gdisk /dev/disk0` (change `/dev/disk0` to `/dev/disk1` or above if your computer has multiple hard disks and the one on which you want to install Ubuntu isn't `/dev/disk0`).
7. In `gdisk`, type `p` to display your partition table. Verify that it's the same one shown by Disk Utility. (Note that Disk Utility hides the EFI System Partition (ESP), which is normally the first partition on the disk.)
8. If you created a Linux partition, type `d` in `gdisk` to delete it. Enter its number when prompted. (If you created multiple partitions for Linux with the intention of using them as you created them, you can skip this step.)
9. In `gdisk`, type `n` to create a new partition. Give it a number of 99, hit the Enter key to use the default starting sector (the program may tell you it's changed the sector value, but you can ignore this message), `+128M` as the ending sector, and a hex code of `ef02`. This creates a BIOS Boot Partition of 128 MiB at the start of the largest block of free space on the disk. This partition will be used by the initial Ubuntu GRUB installation, but you can delete it once you've got native EFI booting working. 128 MiB is actually much bigger than is required for this purpose, but OS X likes to see gaps between partitions when it installs or upgrades the OS, so when you eventually delete this partition, it will leave the type of gap that OS X likes to see. (If you created all your Linux partitions in OS X's Disk Utility, you can put the BIOS Boot Partition in one of the gaps that Disk Utility leaves, but you may need to accept a default value or use `+1M` rather than enter `+128M` as the end value.) Type `p` again to see the results; they should look something like this (although I've got two OS X partitions):



```

Terminal — sudo — 80x24
First sector (34-625142414, default = 469787320) or {+}size{KMGTP}:
Last sector (469787320-625142414, default = 625142414) or {+}size{KMGTP}: +128M

Current type is 'Linux/Windows data'
Hex code or GUID (L to show codes, Enter = 0700): ef02
Changed type of partition to 'BIOS boot partition'

Command (? for help): p
Disk /dev/disk0: 625142448 sectors, 298.1 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): 5A27C85B-AA76-41E5-8D46-59C989B3E1AA
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 625142414
Partitions will be aligned on 8-sector boundaries
Total free space is 155617245 sectors (74.2 GiB)

Number  Start (sector)    End (sector)  Size      Code  Name
   1            40                409639      200.0 MiB EF00  EFI System Partition
   2         409640          78534639    37.3 GiB  AF00  OS X 10.4
   3         78796784         176453039    46.6 GiB  AF00  OS X 10.6
   4        176715184         469787319    139.7 GiB AF00  Shared
   99        469787320         470049463    128.0 MiB EF02  BIOS boot partition

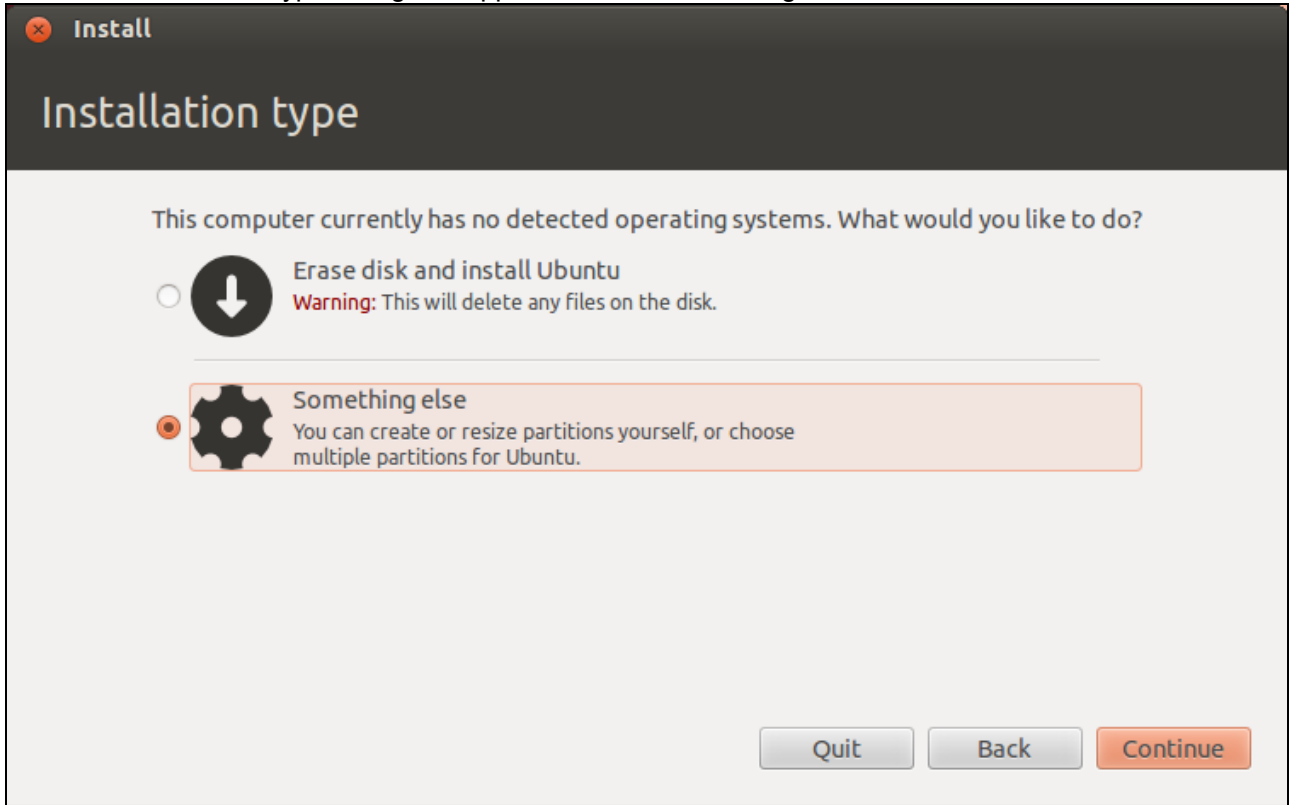
Command (? for help):

```

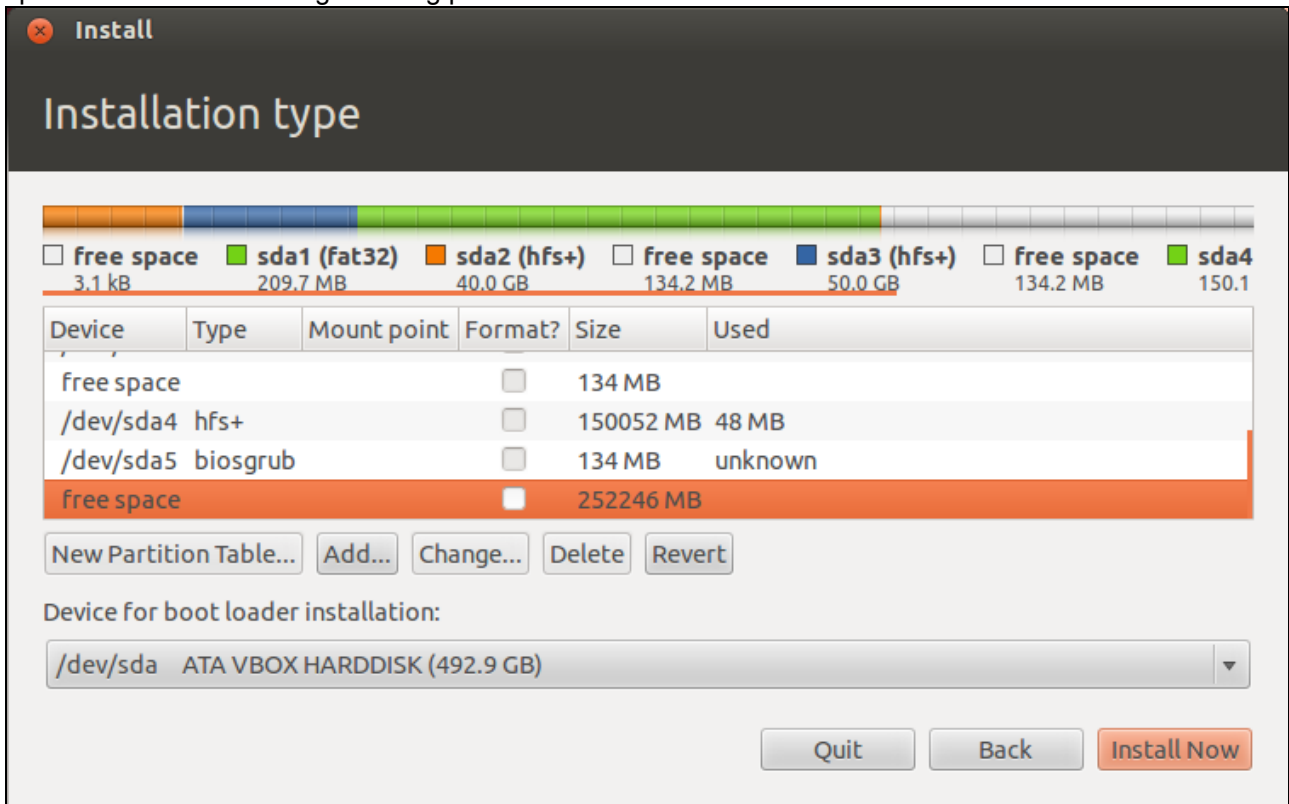
10. In `gdisk`, type `w` to save your changes. When asked, type `y` to confirm this choice.
11. Insert your Ubuntu installation CD or DVD in the computer's drive, or plug in the USB flash drive that holds the Ubuntu installer.
12. Restart your Mac and hold down the Option key (Alt if you're using a PC keyboard) as it restarts to get the Mac's OS selection screen.
13. Pick the Ubuntu install disc from the options. (It may be mis-labelled as "Windows.") The Ubuntu installer will start up. If you're using a CD and the installer doesn't boot, try using a USB flash drive instead.
14. When the appropriate screen comes up, select Install Ubuntu and proceed with a normal Ubuntu

installation, except as noted in the next few steps....

- When the Installation Type dialog box appears, select "Something Else."

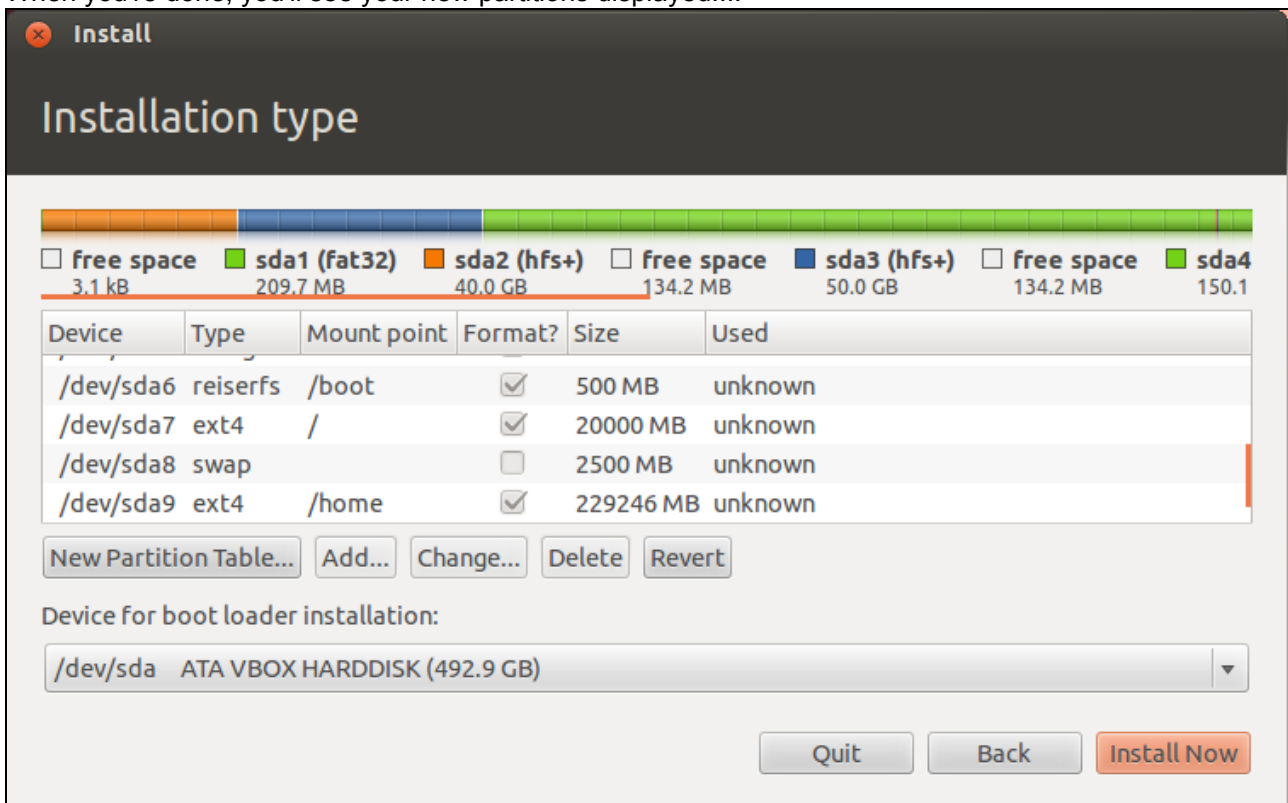


- You will initially see a partitioning screen with your original OS X partitions, a biosgrub partition (what Ubuntu's partitioner calls the BIOS Boot Partition), and free space at the end of the disk. Select the free space and click Add to begin adding partitions....



17. Create partitions for your Ubuntu installation in the free space near the end of the disk. (If you did this in OS X's Disk Utility, you should modify each partition by selecting it and clicking Change, then clicking the Format option and setting an appropriate filesystem and mount point.) I recommend creating the following partitions:
- **/boot**—500 MiB, ReiserFS or ext2fs. This partition is optional. It can simplify some types of boot loader configurations, particularly if you want to use XFS, JFS, or Btrfs on your root (/) partition.
  - **/ (root)**—5–25 GiB, ext4fs or your preferred Linux filesystem. This partition is absolutely vital. If you choose not to create a separate /home partition, make the root partition suitably larger. Take note of the number of this partition (7 in the below screen shot).
  - **swap**—1–2 times your system's RAM size.
  - **/home**—The rest of your available disk space, in ext4fs or your preferred Linux filesystem. You can merge this with the root (/) partition if you're short on space or plan to store little in the way of Ubuntu-specific files. Having a separate /home partition simplifies OS re-installations and some types of upgrades, though.

When you're done, you'll see your new partitions displayed....



18. Be sure that the "Device for boot loader installation" option is set to /dev/sda (it will include a complete name for your hard disk—I used VirtualBox to collect the last few screen shots, but yours will show a real disk model number).
19. Click Install Now. The installation will proceed as on a PC. You can find generic installation guides online, such as [this one](#) at the Ubuntu site.

When the installation finishes, the computer will reboot—straight into Mac OS X! (See the below note if it doesn't boot, though.) You must now proceed with fixing it up in various ways....

## Fixing the Installation

If you've followed the directions, your computer should now be booted into OS X, looking very much like it did before. Ubuntu *is* installed, however, and your disk has a hybrid MBR. You must now take steps to return the hybrid MBR to a safer protective MBR, as the GPT standard requires, and to set up a boot loader that enables you to select which OS to boot when the computer powers up. To do so, follow these steps:

**Note:** If by chance your computer won't boot, try booting the Ubuntu installer but click Try Ubuntu to run it in its live CD mode. Then install the Ubuntu version of GPT fdisk and run steps 1–7 below in the Ubuntu live CD. (Use `/dev/sda` rather than `/dev/disk0`; Linux and OS X use different disk identifiers.) With any luck, OS X will start up again, and you can proceed with step #8.

1. In OS X, launch a Terminal.
2. Type `sudo gdisk /dev/disk0` (changing the disk identifier as necessary). If at any point in the next few steps something seems wrong, type `q` to exit without saving your changes.
3. In `gdisk`, type `x`. The command prompt will change to read `Expert command (? for help):.`
4. In `gdisk`, type `o`. This command displays the contents of the hybrid MBR, which will probably consist of four partitions, one of which is of type `0xEE`. The Ubuntu installer created a hybrid MBR (if one wasn't already present) in an attempt to be helpful.
5. In `gdisk`, type `n`. The program won't seem to do anything; it will just show you another command prompt.
6. In `gdisk`, type `o` again. The MBR contents should be different from before; there should just be one partition, of type `0xEE`. This is a standards-compliant protective MBR.

```

Terminal — sudo — 80x24
Command (? for help): x
Expert command (? for help): o
MBR disk identifier: 0x00000000
MBR partitions:
Number  Boot  Start (sector)  Length (sectors)  Type
  1      1      1                409639            0xEE
  2      2      409640           78125000          0xAF
  3      3      78796784        97656256          0xAF
  4      4      176715184       293072136        0xAF

Disk size is 625142448 sectors (298.1 GiB)

Expert command (? for help): n
Expert command (? for help): o
MBR disk identifier: 0x00000000
MBR partitions:
Number  Boot  Start (sector)  Length (sectors)  Type
  1      1      1                625142447        0xEE

Disk size is 625142448 sectors (298.1 GiB)

Expert command (? for help):

```

7. In `gdisk`, type `w` to save your changes.
8. Unpack the rEFInd zip file you downloaded earlier. If you downloaded the CD image, mount it.
9. In the Terminal, change into the directory created by unpacking the rEFInd zip file, or to the CD image's mount point.
10. Type `./install.sh`. This runs a script that installs rEFInd to your OS X boot partition and "blesses" the program so that the Mac will use it when you next boot. If you're using whole-disk encryption on your OS X partition or if you want to install rEFInd to the ESP rather than to the OS X boot partition, type `./install.sh esp` rather than `./install.sh`.

**Note:** The rEFInd installation instructions presented here are simplified. For the full instructions, including some variants, see [the rEFInd installation page](#).



**Update:** If you're using a 3.3.0 or later kernel, you can skip most of the rest of this page, and instead perform a much simpler operation:

1. Copy an EFI driver for the filesystem you used on `/boot` (or your root filesystem, if you didn't split off `/boot`) from the `rEFInd` package to the `drivers` subdirectory of the `rEFInd` installation directory, which is normally `/EFI/refind`. Note that you must copy the driver for your EFI's architecture—architecture codes appear in the filesystem drivers' filenames. If you did not use `ext2/3/4fs` or `ReiserFS` on `/boot`, this variant procedure will not work.
2. When you reboot, highlight one of the Linux options that refers to a file called `vmlinuz-version`, where `version` is a version number.
3. Press `F2` or `Insert` twice to open a line editor.
4. Add `ro root=/dev/sda5` to the kernel options, changing `/dev/sda5` to your root filesystem's identifier.
5. Press `Enter`. This should boot Linux, although the screen might go completely blank for a few seconds.
6. If it doesn't already exist, create a directory called `/boot/efi`.
7. Edit `/etc/fstab` and add an entry to mount your ESP (normally) `/dev/sda1` at `/boot/efi`. The entry should resemble the following:
 

```
/dev/sda1    /boot/efi    vfat        ro,mask=133    0 0
```
8. Type `mount -a` to mount the ESP at `/boot/efi`.
9. Run the `mkrlconf.sh` script that comes with `rEFInd`, as in `sudo ./mkrlconf.sh`, typed from the directory where the file exists. This action should create a file called `/boot/refind_linux.conf`, which is briefly described in the [Improving the Boot Method](#) section.

At this point, it should be possible to boot Linux by rebooting the computer and selecting one of the `vmlinuz-version` entries in `rEFInd`'s menu. If this doesn't work, continue with the main procedure described here....

11. Load `/efi/refind/refind.conf` into a text editor and locate the commented-out `scanfor` line. (If you installed `rEFInd` to the ESP, you'll need to mount the ESP and load the configuration file from there.) Uncomment the `scanfor` line by removing the leading hash mark (`#`) and then add a new item, `cd`, to the end of the line. It should read `scanfor internal,external,optical,cd`. (If you're using USB flash drives, add `bioexternal` instead of or in addition to `cd`.)
12. Reboot. The `rEFInd` menu should appear, as shown below. Your menu options might differ from these, though. (This system has two OS X installations and a working OpenSUSE installation accessible via two entries.)



13. Insert the Super GRUB 2 Disk into the computer's DVD drive.
14. Restart your Mac by selecting the reboot item (the yellow icon with the circular arrow on the far right of the row of smaller icons).
15. When the rEFInd menu re-appears, you'll see a new Linux icon on the far right of the display. (Depending on your screen's size and the number of OS loaders rEFInd discovers, you may need to scroll over to see it.) Select it.
16. After a while, a GRUB 2 menu should appear. Pick the `Detect any GRUB 2 installation (even if MBR is overwritten)` option from the menu. Note: On my system, if I leave this menu up for more than two or three seconds, it hangs at this point. If yours is the same, you'll need to act quickly!
17. A new menu should appear listing the GRUB 2 installations it could locate. On my system, this consisted of a grand total of one installation, identified as `(hd1,gpt6)/grub/core.img`, so there's no ambiguity: Select it! (Of course, your disk device numbers are likely to differ from mine.)
18. At this point, a normal Ubuntu GRUB 2 menu should appear, enabling you to boot Ubuntu as you would on a PC. (The screen may go completely black for part of the boot process. Don't worry; this is normal.)

```

GNU GRUB  version 1.99-21ubuntu3

Ubuntu, with Linux 3.2.0-23-generic-pae
Ubuntu, with Linux 3.2.0-23-generic-pae (recovery mode)
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)
Mac OS X (32-bit) on /dev/sda2
Mac OS X (64-bit) on /dev/sda2
Mac OS X (32-bit) on /dev/sda3
Mac OS X (64-bit) on /dev/sda3

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.

```

19. When the login screen appears, log into your Ubuntu installation.
20. Click Dash Home (the icon in the upper-left corner of the screen) and type **term** in the search field. A few icons should appear, including one called Terminal. Click it to launch a Terminal program.
21. Type **sudo mkdir /boot/efi** to create the standard mount point for the EFI System Partition (ESP).
22. Type **sudo mount /dev/sda1 /boot/efi** to mount the ESP at /boot/efi. (Change /dev/sda1 if your ESP has an unusual partition number.) You can use `gdisk` to check the ESP's number, if you like.
23. Type **ls /boot/efi**. You should see the contents of the ESP, which will probably consist of a single directory called `efi`. There could be other files and directories, but probably not many of them. If it appears you've mounted the wrong partition, review your partition layout and the commands you've typed to mount the ESP. Proceed only when you're sure you've mounted the ESP at /boot/efi.
24. Type **sudo apt-get install grub-efi**. This replaces the BIOS version of GRUB that the Ubuntu installer installed with an EFI-based version of GRUB. In fact, there are two different EFI-enabled versions of GRUB: `grub-efi-ia32` and `grub-efi-amd64`. Installing `grub-efi` installs the package that's appropriate for your system, assuming you installed the optimum architecture. If you're running a 32-bit version of Ubuntu on a recent 64-bit OS X firmware, you may need to explicitly install `grub-efi-amd64`. Installing `grub-efi` also installs a number of dependencies, such as `efibootmgr`.
25. Type **sudo mkdir /boot/efi/efi/ubuntu/**. This command creates a home for GRUB on the ESP. (Note the doubled-up `efi` in the path. This is *not* a typo.)
26. Type **sudo grub-install**. (Recent versions of GRUB seem to require an option. Pass it your ESP's device filename, as in **sudo grub-install /dev/sda1**.) You'll see a few messages appear on the screen, including a couple of "fatal" errors about EFI variables. Ignore those messages. They're caused by the fact that you're booted in BIOS mode, and they're irrelevant because the task they're intended to perform will be handled by `rEFInd`.
27. Type **ls -l /boot/efi/efi/ubuntu**. You should see two files, `boot.efi` and either `grubia32.efi` or `grubx64.efi`, depending on your platform. They should have the same file size; in fact, they're identical.
28. If both of those files are present, remove one of them; having both will just clutter your `rEFInd` menu

unnecessarily. If you somehow wound up with just one file, though, leave it in place. If you don't see either file, then do some troubleshooting now; you won't be able to boot into Ubuntu without one of those files (or some other EFI boot loader for Linux).

At this point, if you did everything exactly correctly, you should be able to boot Ubuntu in EFI mode. When you reboot, your rEFInd menu should include a new Ubuntu option, as shown below. Select it and your GRUB menu should appear; it will resemble the one shown earlier, although it may use a different font and color scheme.



Once you're satisfied with your ability to boot and use both Linux and OS X, you can delete the BIOS Boot Partition from your hard disk. It's no longer needed, but OS X may want free space where it resides in the future. You can use GParted, parted, gdisk, or any other partitioning tool to delete this partition.

Although my own system doesn't seem to suffer from its presence, it's conceivable that some Macs will experience boot-time slowdowns because of the presence of the BIOS version of GRUB's boot code in the hard disk's MBR. If you think this is happening, you can type `sudo dd if=/dev/zero of=/dev/sda bs=440 count=1` to eliminate it. Be *very careful* with that command, though! Be *absolutely positive* that you've typed it correctly, and particularly the `bs=440` and `count=1` numbers. If you write too much data in this way, you can damage your partition table!

If you've not used it before, you may want to peruse the [rEFInd documentation](#). Although the default options work well for most systems, you may want to tweak some of them or install ancillary programs, such as an EFI shell program.

You may want to add an entry for the ESP to your `/etc/fstab` file so that it will be mounted automatically whenever you boot. The following line will do the trick on most systems:

<code>/dev/sda1</code>	<code>/boot/efi</code>	<code>vfat</code>	<code>ro, fmask=133</code>	<code>0 0</code>
------------------------	------------------------	-------------------	----------------------------	------------------

You can tweak this entry as you see fit. The `/dev/sda1` specification works for most people, but you could change it to use a LABEL or UUID specification, as in `UUID=2B68-9A85`. This will make the configuration more robust should the disk identifier change because you boot with a different disk configuration or you repartition the disk. If you this change, you'll need to obtain the label or UUID value for your ESP. Typing `blkid /dev/sda1` (changing the device identifier, if necessary) should do this.

## Improving the Boot Method

Although GRUB 2 often works in a configuration like the one I've just described, it also often misbehaves. For instance, on my brand-new Ubuntu 12.04 installation (less than 24 hours old as I type), GRUB 2 often gives me a "file not found" error when selecting the Ubuntu entry; then if I select the *same entry*, it boots fine. On other installations and computers, GRUB 2 has resulted in kernels that begin to boot and then fail; it's begun returning a `grub>` emergency prompt for no apparent reason; and it's done a poor job of detecting OSES and kernels. What's more, on my Mac, Ubuntu's packaging system keeps trying to replace `grub-efi` with `grub-pc`, which would result in an unbootable installation if allowed to continue! For these reasons, GRUB 2 is the EFI boot loader that I favor *least*. That said, some people say it works acceptably for them, and it's Ubuntu's default boot loader, so Ubuntu's installation scripts usually do a tolerable to good job of automatically configuring it. These are the reasons I provided instructions for installing GRUB 2. I personally try to remove it as soon as possible after an installation, or at least shove it out of the way so that it can't do any real damage.

If you have problems with it, you can switch to another EFI boot loader. My [EFI Boot Loaders for Linux](#) Web page describes the options that are available. If you've followed the directions on this page, my personal favorite is actually quite close to working: If you can install a Linux kernel with the [EFI stub loader](#) support enabled, rEFInd can launch it directly, without the help of GRUB. The major hurdle to overcome on this score is that Ubuntu 12.04 installs a 3.2.0 kernel, but the EFI stub loader became available with the 3.3.0 kernel. This is why I recommended installing Ubuntu 12.04.2, 13.04, or a more recent version of Ubuntu; these versions ship with 3.3.0 or later kernels. In broad outline, to use the EFI stub loader, you must:

1. Obtain a kernel with an EFI stub loader, either by finding a suitable binary package or by compiling the kernel yourself.
2. Optionally install an [EFI driver](#) for the filesystem you used on your Linux `/boot` partition. This isn't required, but it can simplify setup by enabling the EFI to load the kernel from its normal home in `/boot`. If you omit this step, you'll need to copy the kernel to a FAT or HFS+ volume. (The possibility of using this approach to booting is why I recommended creating a separate ReiserFS or ext2fs `/boot` partition; those are the Linux filesystems for which EFI drivers are available.)
3. Install the new kernel and initial RAM disk in a place where the ESP can read it. As just noted, this can be `/boot` if you've installed an EFI driver. If not, the ESP can work, at least in theory (it doesn't on my Mac Mini, but that may be a model-specific problem). An HFS+ volume is also an option. (Odd as it may sound, converting `/boot` to use HFS+ can work quite well on Macs—but not on UEFI-based PCs.)
4. Either rename the Linux kernel file so that it includes a `.efi` extension or uncomment the `scan_all_linux_kernels` option in rEFInd's `refind.conf` file. The former method works best if you have kernels that do *not* have EFI stub loader support enabled in directories that rEFInd will scan. The latter method works well if you're using a distribution that provides suitable kernels and you've loaded drivers so that the EFI can read `/boot`.
5. Create a file called `refind_linux.conf` in the directory that holds your kernels. This file holds kernel option sets, one per line, as in:

```
"Boot using standard options" "root=UUID=17c62fb9-a67a-4bae-97d7-3b71e5d3c8ce ro
"Boot using minimal options" "root=UUID=17c62fb9-a67a-4bae-97d7-3b71e5d3c8ce ro"
"Boot using recovery options" "root=UUID=17c62fb9-a67a-4bae-97d7-3b71e5d3c8ce ro re
```

This `refind_linux.conf` file is taken from a working installation, but you'll need to adjust the UUID value for your own computer if you use it as a model. When you boot, rEFInd uses the first line's options by default, but by pressing F2 or Insert when you select a kernel in rEFInd, you can choose another option set.

The beauty of this method of booting is that you don't need to change configuration files when you install a new kernel—just place it, with a suitable filename, in the kernel directory and rEFInd will detect it and use the options from `refind_linux.conf` to boot it. Although GRUB 2 is supposed to be as easy to use, it accomplishes this goal by running a series of configuration scripts whenever you install a new kernel and updating its configuration file. This process can go wrong, particularly in advanced or exotic setups.

If you have problems with GRUB 2, ELILO and GRUB Legacy are the best options, as described on my [EFI](#)

[boot loaders](#) page. Each has advantages and disadvantages. On my Mac Mini, ELILO doesn't work, but that may be a model-specific quirk. Unfortunately, you'll need to adjust either boot loader's configuration whenever you upgrade your kernel, so keep an eye on that!

## Going Further

You can consult the [UEFIBooting](#) page to learn about more options for GRUB on a Mac. Some of that page's instructions didn't work for me, though; presumably they're either specific to particular Mac models or the page is dated.

I hope you've found this Web page informative. I realize the procedures outlined here are complex, but in my opinion, it's worth getting rid of the flaky and dangerous hybrid MBR even if doing so requires investing the effort described here.

## References

The following sites have additional information that you might find useful:

- The [Linux on Your Apple Mac wiki](#) has a variety of information on Linux/Mac integration.
- The [Arch Linux on a Macbook Air 4.2](#) site describes an EFI-only installation of Arch Linux on a Mac.
- I guess Arch people like Macbooks, since there's [another site](#) with instructions on installing Arch to a Macbook Air using an EFI boot.
- The [Booting Linux Inside a MacBook Using GRUB and EFI](#) page outlines the process with an emphasis on Debian and Ubuntu.
- My [EFI Boot Loaders for Linux](#) page describes all of the major Linux boot loaders and boot managers for EFI-based computers.
- [My Linux on UEFI: A Quick Installation Guide](#) page provides helpful tips on how to install Linux on EFI-based systems.
- You can learn about additional tweaks for rEFInd at [its official documentation site](#).

---

If you have problems with or comments about this web page, please e-mail me at [rodsmith@rodsbooks.com](mailto:rodsmith@rodsbooks.com). Thanks.

[Return](#) to my main Web page.