

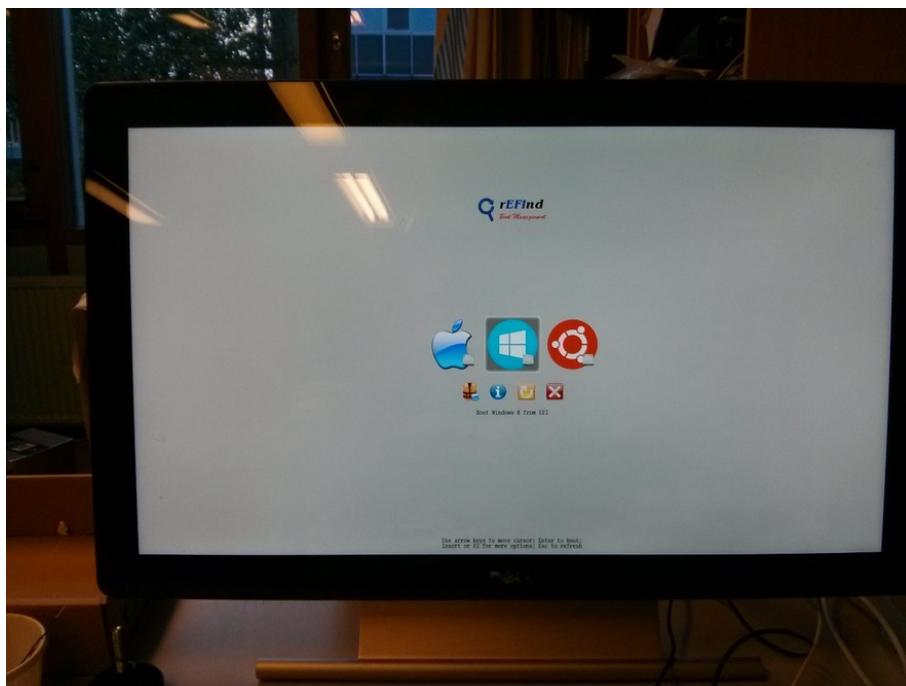
# Triplebooting Macintosh with OS X, Windows, and Ubuntu

## Purpose

---

This documentation serves to help you install Windows and Ubuntu, alongside an existing OS X installation, on a Macintosh newer than 2013 or so.

A completed setup should boot into a screen as the one displayed below, allowing the user to select which OS they wish to continue to boot into



## Prerequisites

---

- Macintosh - With OS X installed
- Ubuntu ISO file (64 bit version)
- Windows ISO file (64 bit)
- refind - Get it from [here](#) (I recommend the binary zip file method)
- USB storage device or DVD - Large enough for Ubuntu / Windows installation (8 GB or so)

This process should work with any newer versions of the above mentioned hardware and software, however here are the exact specifications tested:

- Macmini7,1 (Late 2014 mid range Mac Mini)
- OS X 10.10.5
- Ubuntu 14.04.3
- Windows 8 Professional
- refind 0.9.2
- 2xKingston DTSE9 G2 16 GB

On top of the physical requirements, the guide presumes you have at least a basic understanding of disks, partitions, and using the commandline.

## Overview

---

The installation process will, largely consist of:

1. Resize the OS X partition to make space for the other systems
2. Boot Ubuntu live and create partitions for Windows and Ubuntu with GParted
3. Install Ubuntu
4. Fix partition table (Ubuntu will alter this during it's installation, in a way we do not want)
5. Install Windows and drivers
6. Install and configure rEFInd (The bootmanager that will let us select OS)
7. **Install Ubuntu drivers and updates** (Needs to be documented) see  
<https://help.ubuntu.com/community/WifiDocs/Driver/bcm43>  
XX
8. **Install Windows drivers and updates** (Needs to be documented)

Ready for the details? Continue below!

## Process

---

### Resizing the OS X partition

Assuming you haven't done anything special to your OS X installation, it should be taking up the entirety of the hard drive

in the Mac. So the first step will be to resize OS X, to make space for the two other operating systems.

You may choose any amount of space for each operating system, as long as it meets the requirements of the specific version of the OS you are using.

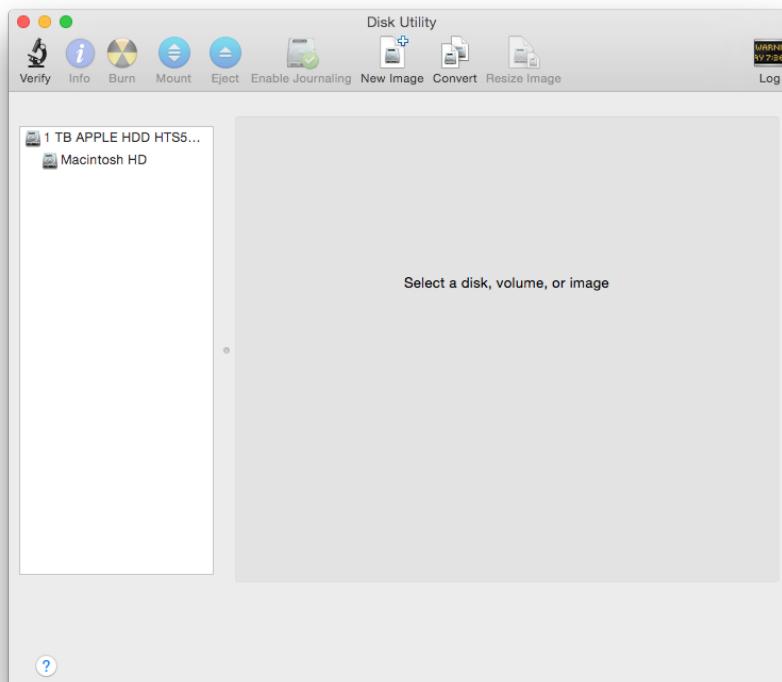
To make things simple, I will be splitting my hard drive in 3 roughly equal bits.

As my drive is 1 TB large, I will be resizing OS X to 330 GB, leaving me 670 GB for the rest.

**Note** whichever amount of space you choose to leave to OS X, it must be at least the amount of space the system is currently using. OS X might either deny you the right to make it as small as you're trying, or (worse) simply overwrite some of the files that are already on your system. If you want the OS X partition to be smaller than the amount of space it's currently using, clear some space in OS X first, before resizing the partition.

On the Mac, open *Disk Utility*, either by searching in spotlight, or navigating to *Applications > Utilities > Disk Utility* using *Finder*.

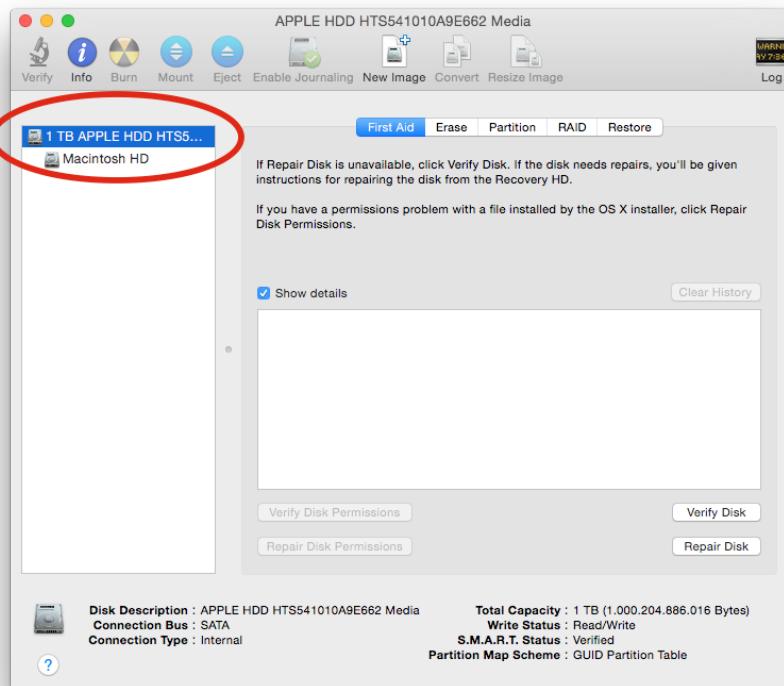
You should be met with a view like this:



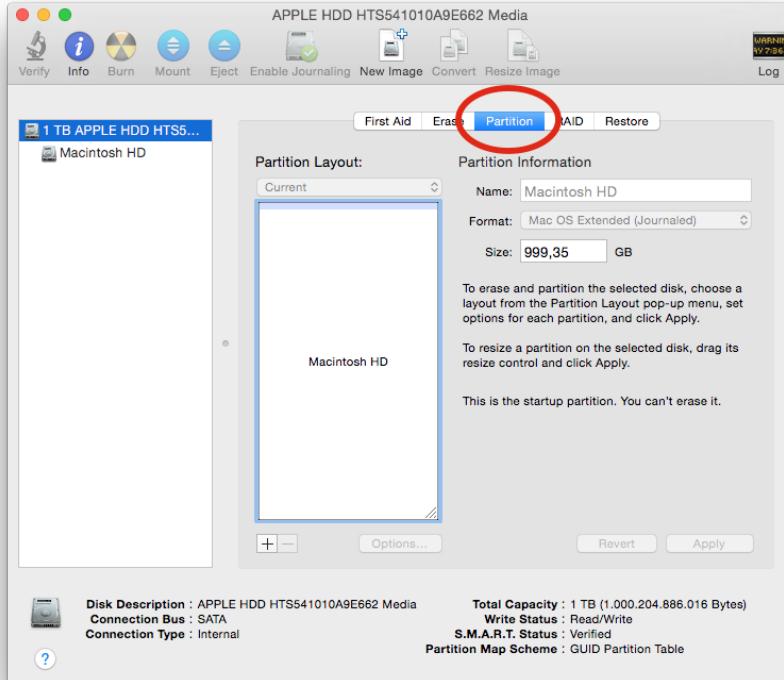
On the far left side, you see a list of storage devices connected to the computer, and the partitions on them.

In my case, I only have the internal hard disk of the computer available, with my single standard OS X partition. You may have multiple devices available, with a variable number of partitions on them.

Select the device on which your OS is installed, and on which we will be installing the others, by clicking on it. (In my case the 1TB Internal Apple drive)



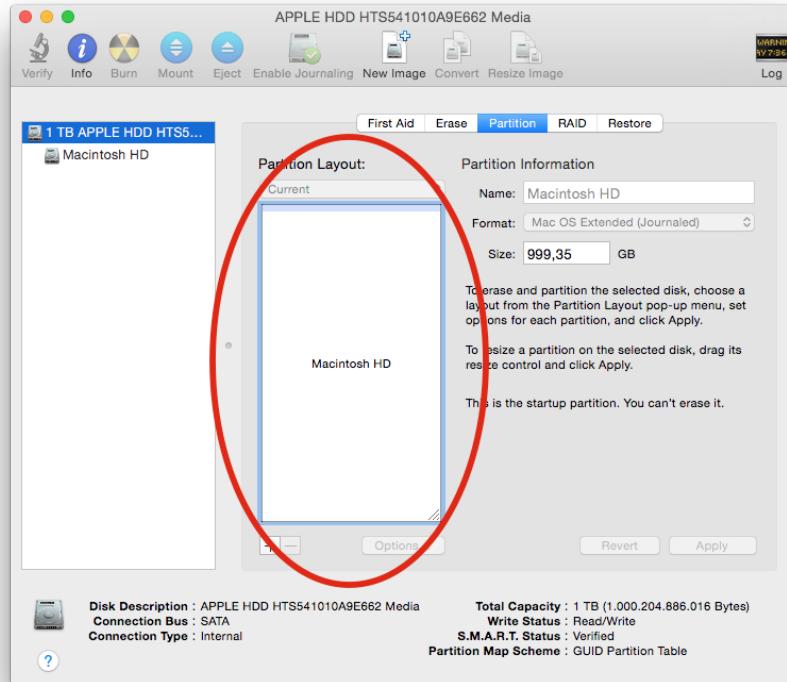
You then need to select the *Partition* view, so that we can change the size of the OS X partition.



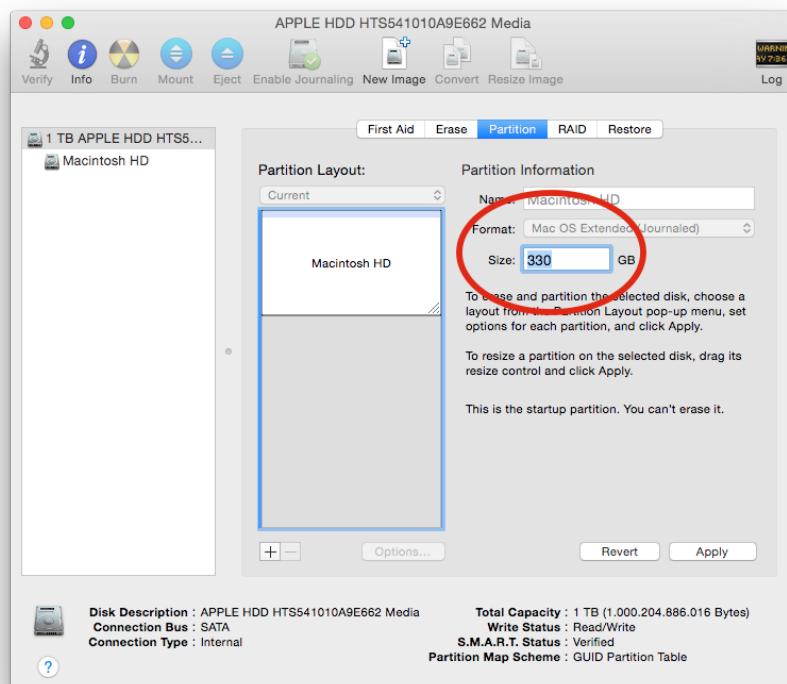
You should then be met with a screen similar to the one above.

**Note** that if you have made any other partitions, or resized the OS X partition, the view might be different. In that case, you will want to select any partitions you have, click the - (Minus symbol) in the lower left of the view, and confirm deletion of the partitions. Once you are left with only the OS X partition, you can continue, disregarding the fact that my partition will be starting out as larger than yours.

The OS X partition should automatically be marked as selected, highlighting it with a blue border, as seen below.



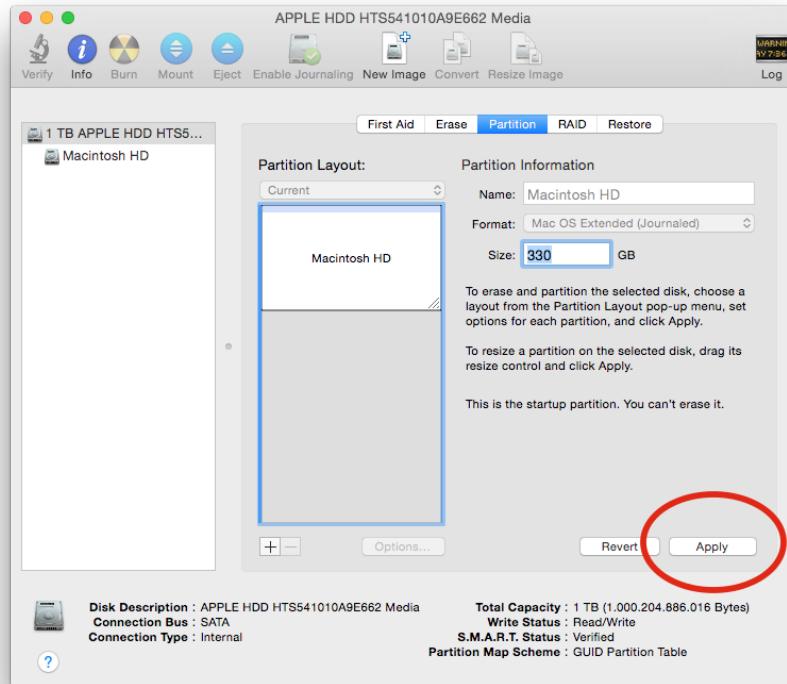
Otherwise you can simply click on it yourself in this view. Click the size box in the right hand side of the view, and enter the new size you wish to set for the partition, and click *enter* on your keyboard. As mentioned earlier, I will be using 330 GB.



The size of the partition in the center view, should change to reflect it's new size.

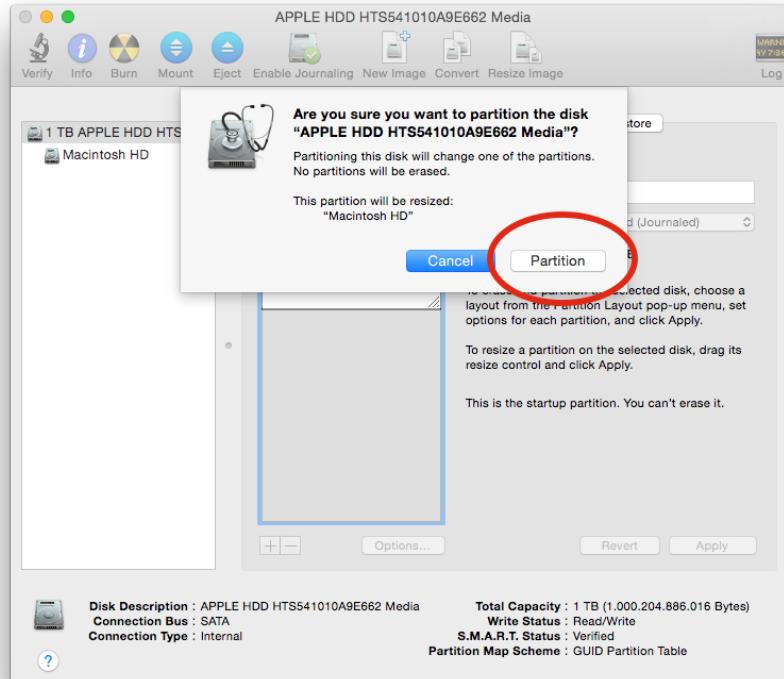
The gray area indicates the empty space that will be left behind, this is where we'll be putting our other two operating systems.

When you're satisfied that the new size is correct, click *apply* in the lower right corner.

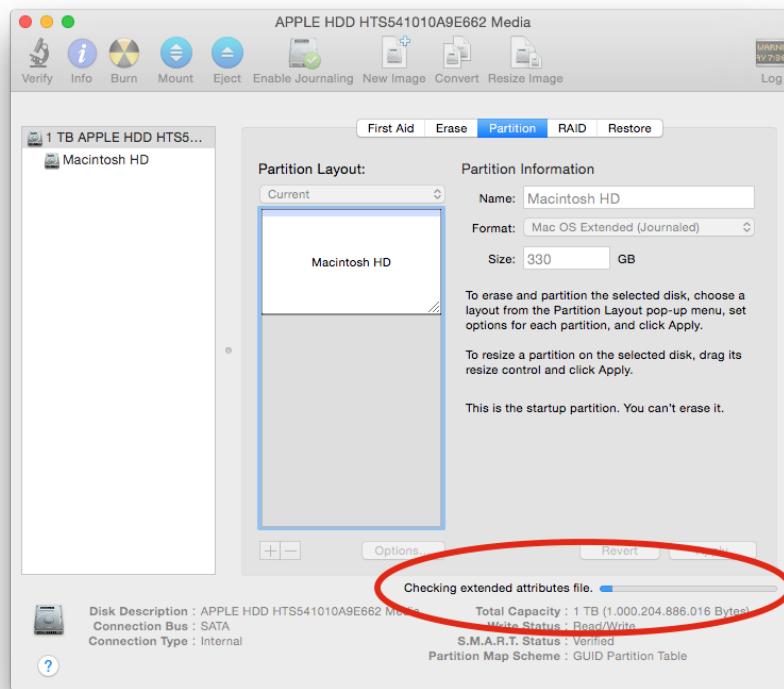


You may be prompted to confirm that you wish to change the size of the partition.

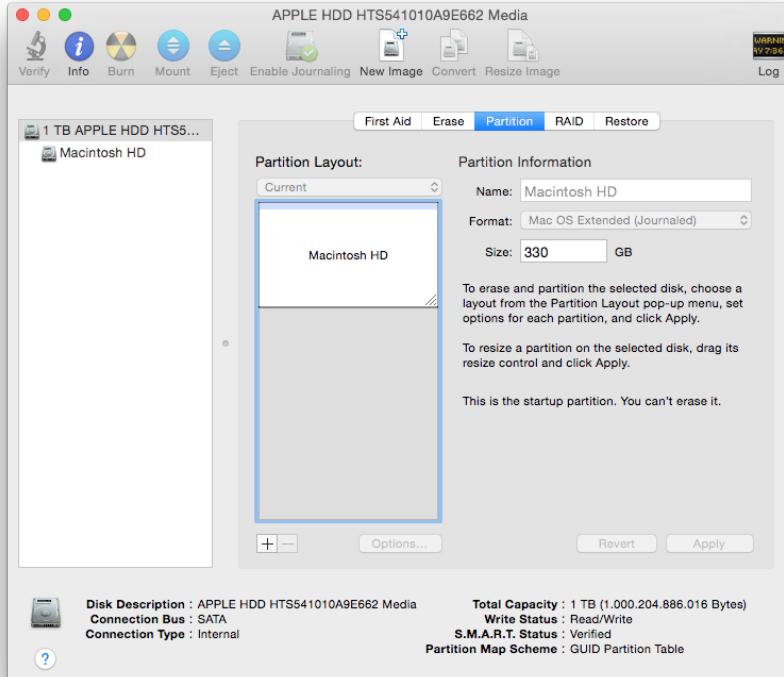
If you are met with this prompt, click *partition* to confirm.



This process may take up to a few minutes. You should see a progress bar in the lower right hand corner, indicating the progress of this process.



Once the process is done, and the progress bar is gone, the partition has been resized.  
 Congratulations, we're now ready to embark on the task of cramming two other systems onto the space we just emptied!



At this point, we are ready to create the new partitions for our two other operating systems.

This could be done from Disk Utility as well, however it does not support the file systems used by Windows and Ubuntu, so we would need to create partitions in Disk Utility, to then replace them with the correct ones later.

Instead, we'll make partitions of the right type, from the start.

This requires us to use GParted, a linux program similar to Disk Utility, but with support for all the file systems we'll need.

## Creating an Ubuntu boot media

We'll need to install Ubuntu onto some media, so that we can boot our Macintosh into Ubuntu, and use GParted. This could be done from a DVD just as well as a USB drive, however the 2014 model Mac Mini has no DVD drive, and a USB drive was more readily available to me, than an external DVD drive.

In either case, excellent documentation is available at [ubuntu.com](http://ubuntu.com)

- [Creating a bootable Ubuntu USB on Windows](#)
- [Creating a bootable Ubuntu USB on Mac](#)

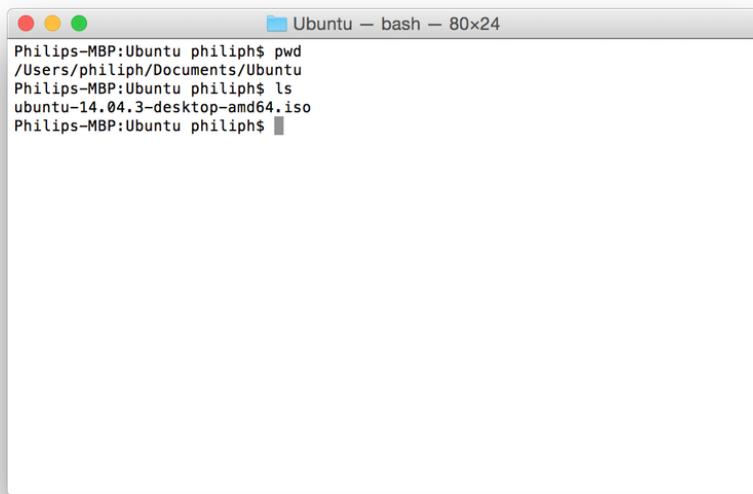
Guides on DVD media can also be found, with a little Google foo.

I'm using the method detailed for Mac, however you can use any other way, as long as it results in ubuntu being written to a media which you can then boot the Mac from.

**Note** all the steps taken here, heavily depend on the exact configuration of your machine, placement of files, and similar. To follow this section you should be comfortable with the command line, and aware of where you might need to split off from my actions.

If you need more detailed, in depth, walkthrough of how to write ubuntu to a USB drive, look online.

As you can see, I have a directory called *Ubuntu* containing only the .iso file downloaded from  
<http://www.ubuntu.com/download>



```
Philip-MBP:Ubuntu philiph$ pwd
/Users/philiph/Documents/Ubuntu
Philip-MBP:Ubuntu philiph$ ls
ubuntu-14.04.3-desktop-amd64.iso
Philip-MBP:Ubuntu philiph$
```

I run the *hdiutil* command to convert the .iso to a .img file, which the mac can write for me.

Given the following command, I enter my values, and execute the command.

```
hdiutil convert -format UDRW -o ~/path/to/target.img ~/path/to/ubuntu.iso
```

```
Philip-MBP:Ubuntu philiph$ pwd  
/Users/philiph/Documents/Ubuntu  
Philip-MBP:Ubuntu philiph$ ls  
ubuntu-14.04.3-desktop-amd64.iso  
Philip-MBP:Ubuntu philiph$ hdiutil convert -format UDRW -o ubuntu-14.04.3-deskt  
op-amd64.img ubuntu-14.04.3-desktop-amd64.iso  
Reading Driver Descriptor Map (DDM : 0)...  
Reading Ubuntu 14.04.3 LTS amd64          (Apple_ISO : 1)...  
Reading Apple (Apple_partition_map : 2)...  
Reading Ubuntu 14.04.3 LTS amd64          (Apple_ISO : 3)...  
*****  
Reading EFI (Apple_HFS : 4)...  
*****  
Reading Ubuntu 14.04.3 LTS amd64          (Apple_ISO : 5)...  
*****  
Elapsed Time: 4.704s  
Speed: 213.8Mbytes/sec  
Savings: 0.0%  
created: /Users/philiph/Documents/Ubuntu/ubuntu-14.04.3-desktop-amd64.img.dmg  
Philip-MBP:Ubuntu philiph$
```

With the image ready to be writing to my USB drive, I must identify which drive I want to write to. Before inserting my USB drive into the computer, I run the command

diskutil list

This gives me a list of disk devices available on my computer.

```
Ubuntu - bash - 80x24
Reading Ubuntu 14.04.3 LTS amd64      (Apple_ISO : 3)...
-----
Reading EFI (Apple_HFS : 4)...
-----
Reading Ubuntu 14.04.3 LTS amd64      (Apple_ISO : 5)...
-----
Elapsed Time: 4.704s
Speed: 213.8Mbytes/sec
Savings: 0.0%
created: /Users/philiph/Documents/Ubuntu/ubuntu-14.04.3-desktop-amd64.img.dmg
Philips-MBP:Ubuntu philiph$ diskutil list
/dev/disk0
 #:          TYPE NAME      SIZE IDENTIFIER
 0: GUID_partition_scheme *251.0 GB disk0
   #:          NAME      SIZE IDENTIFIER
 1: EFI       EFI      209.7 MB disk0s1
 2: Apple_CoreStorage    250.1 GB disk0s2
 3: Apple_Boot Recovery HD   650.0 MB disk0s3
/dev/disk1
 #:          TYPE NAME      SIZE IDENTIFIER
 0: Apple_HFS Macintosh HD   *249.8 GB disk1
   #:          NAME      SIZE IDENTIFIER
   #:          Logical Volume on disk0s2
   #:          3EC93AB6-57AA-4BD7-B095-094C229A4718
   #:          Unlocked Encrypted
Philips-MBP:Ubuntu philiph$
```

I can then enter the USB drive, and observe which new devices are available since.

```

Philips-MBP:Ubuntu philiphs$ diskutil list
/dev/disk0
#:          TYPE NAME      SIZE IDENTIFIER
0: GUID_partition_scheme          *251.0 GB disk0
   #:              EFI  EFI           209.7 MB disk0s1
   2:     Apple_CoreStorage        250.1 GB disk0s2
   3:       Apple_Boot Recovery HD   650.0 MB disk0s3
/dev/disk1
#:          TYPE NAME      SIZE IDENTIFIER
0:     Apple_HFS Macintosh HD    *249.8 GB disk1
   Logical Volume on disk0s2
   3EC93AB6-57AA-4BD7-B095-094C229A4718
   Unlocked Encrypted
/dev/disk3
#:          TYPE NAME      SIZE IDENTIFIER
0: GUID_partition_scheme          *15.7 GB disk3
   1:   Microsoft Basic Data UBUNTU   15.7 GB disk3s1
Philips-MBP:Ubuntu philiphs$ 

```

As disk 0 and 1 are available without the USB drive, and 0, 1 and 3 are available when the drive is plugged in, `/dev/disk3` must be the identifier of my USB drive.

**Note** this is likely to differ on your machine, run the commands and check yourself!

With that knowledge, I can unmount the disk, and start writing the Ubuntu image to it.

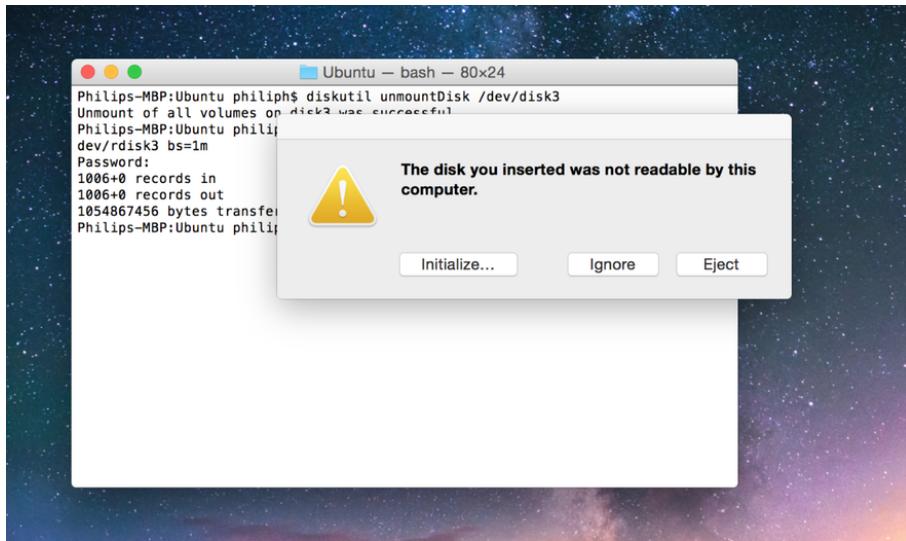
```

diskutil unmountDisk /dev/disk3
sudo dd if=ubuntu-14.04.3-desktop-amd64.img.dmg of=/dev/rdisk3
bs=1m

```

**Note** the r in front of disk, when running the dd command.

The dd command may take up to 10 minutes or so, to write the image to the disk, depending on your hardware. Once it is done, you may be met with a prompt similar to the one below:



In that case, click *Eject*, if not, use *Finder* to find the device, and eject it manually.

We are now ready to utilise the Ubuntu USB, with GParted, to create the partitions we will need to install our systems.

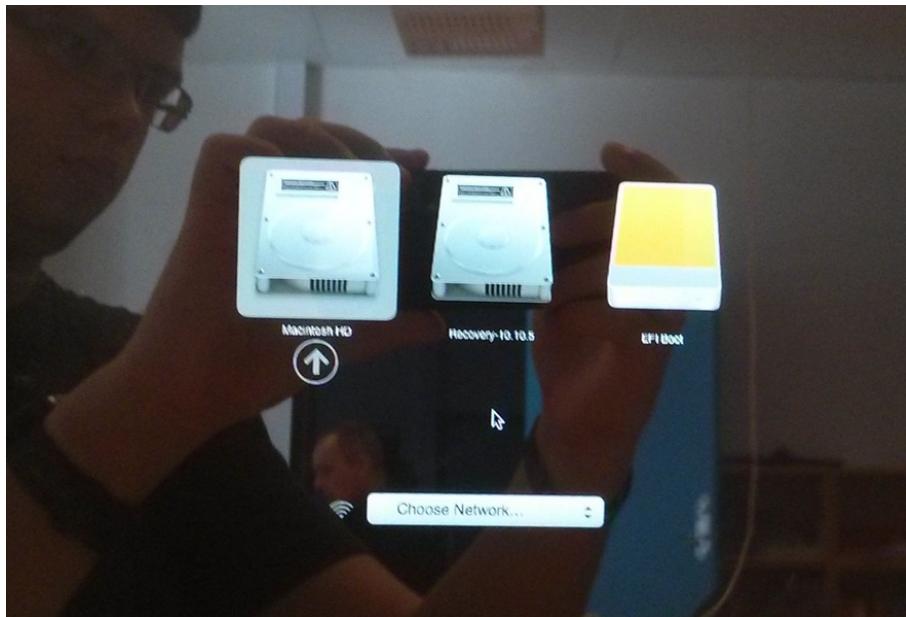
## Using Ubuntu and GParted to create partitions

Once you have an ubuntu media to boot, be it USB or DVD, you will want to start your mac into the ubuntu system.

Insert the media in your mac, and restart it. During the boot, ensure to hold the option key (By default the same key known as alt).

If you do so correctly, you should be met with a screen similar to the one below.

Use the arrow keys and enter, to select the correct boot media for ubuntu (In my case the EFI Boot option - for the USB drive).



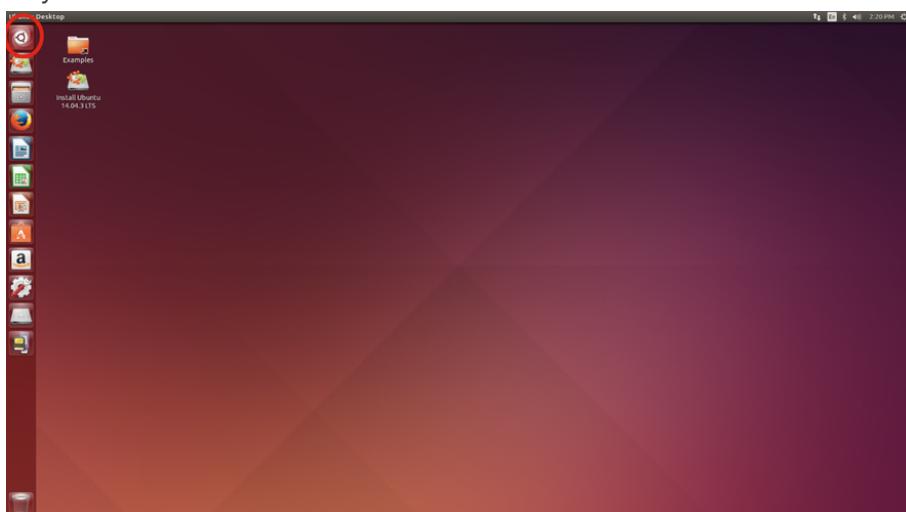
You should then be met with GRUB, the bootloader bundled with Ubuntu. Here you will want to select *Try Ubuntu without installing*.



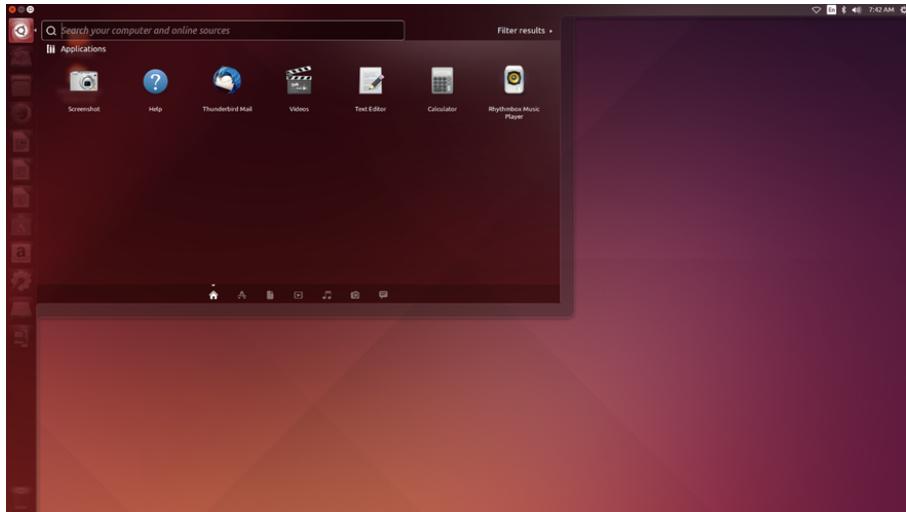
After a little while Ubuntu should be booted up.

Once it is, we're going to use GParted, a program included with the Ubuntu live version, to create three new partitions.

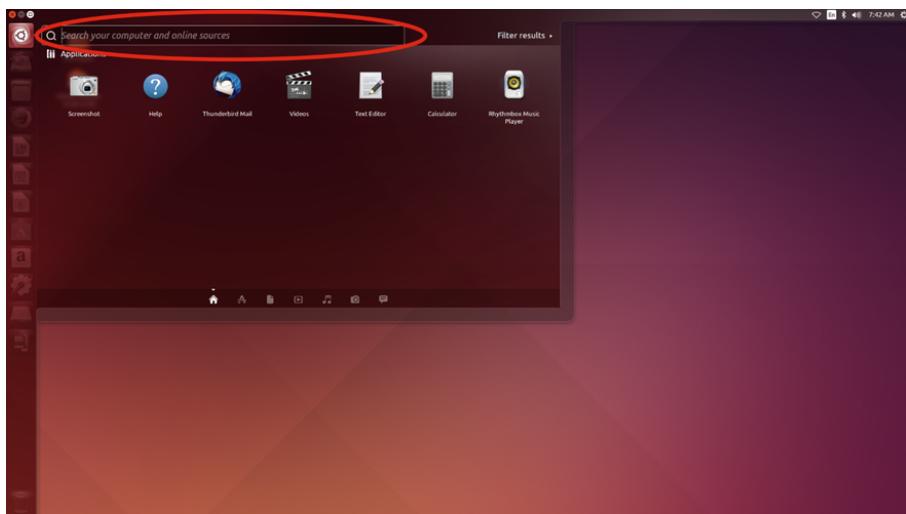
Open the main menu in Ubuntu, either by clicking the logo in the top left corner of the desktop, or by clicking the *Windows* key on a Windows keyboard, or *Command* on a Macintosh keyboard.



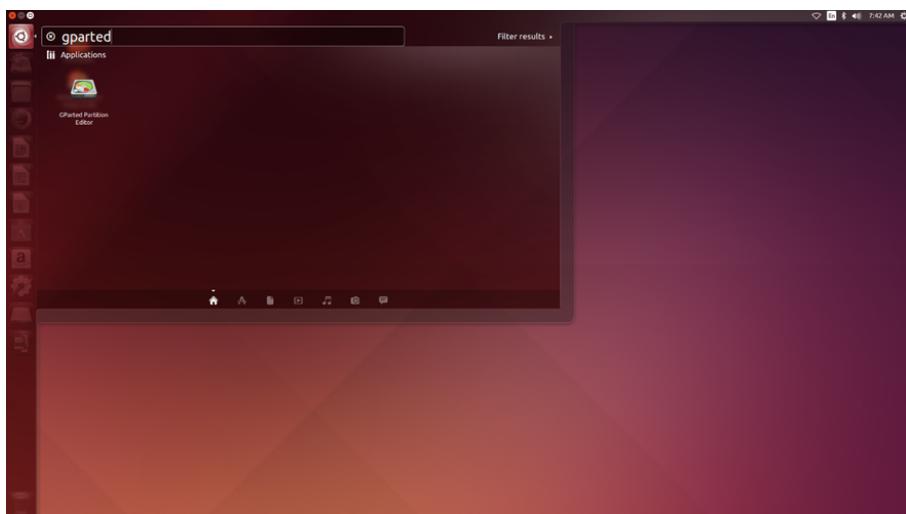
This will bring up the menu



In the menu, select the search field. We will be using this to find GParted

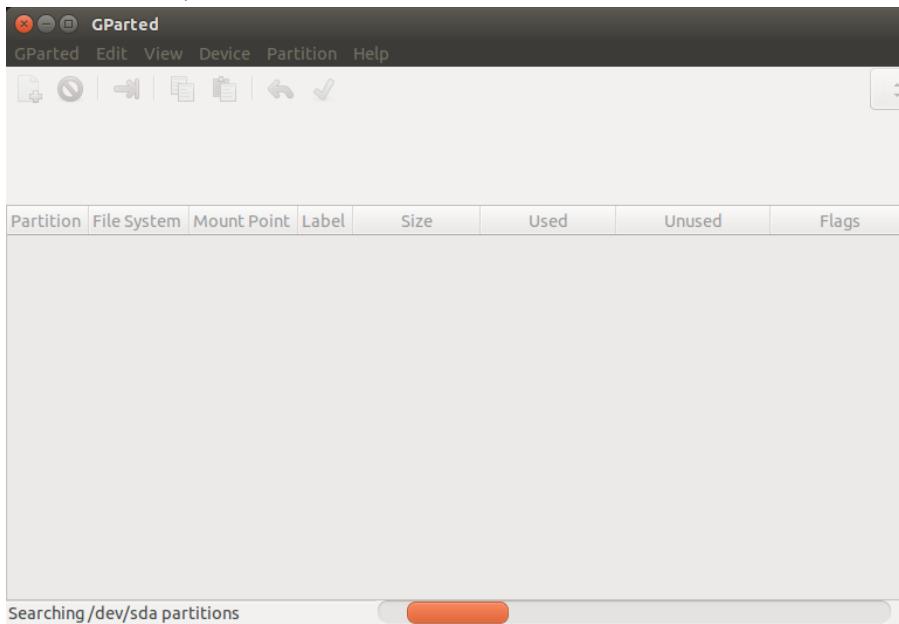


Search for *GParted*, this should only return one result *GParted Partition Editor*.



This is the program we want to start up, either by clicking on it, or navigating to it using the arrow keys and pressing enter. The

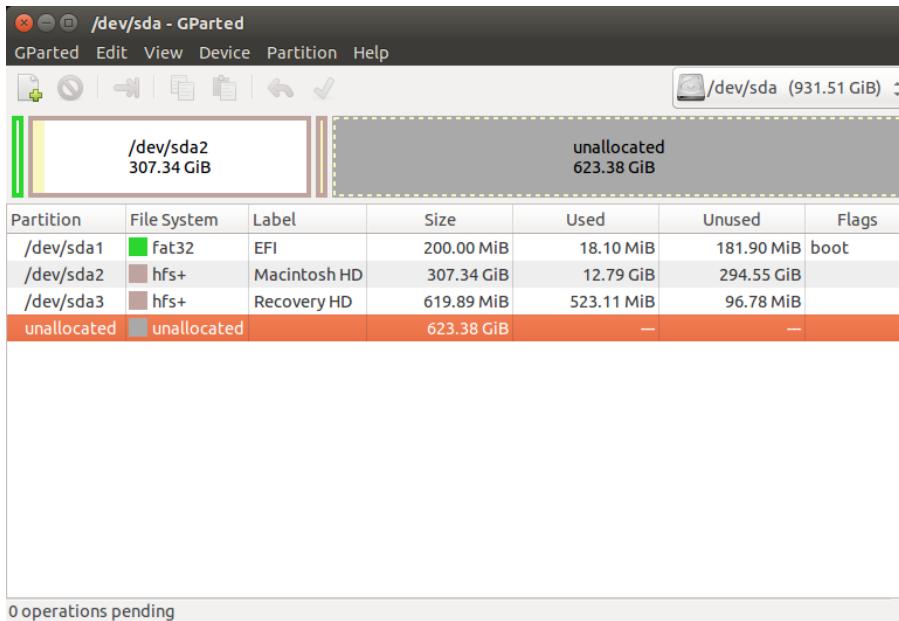
program will take a moment to start up, and scan the first disk it finds for a partition table.



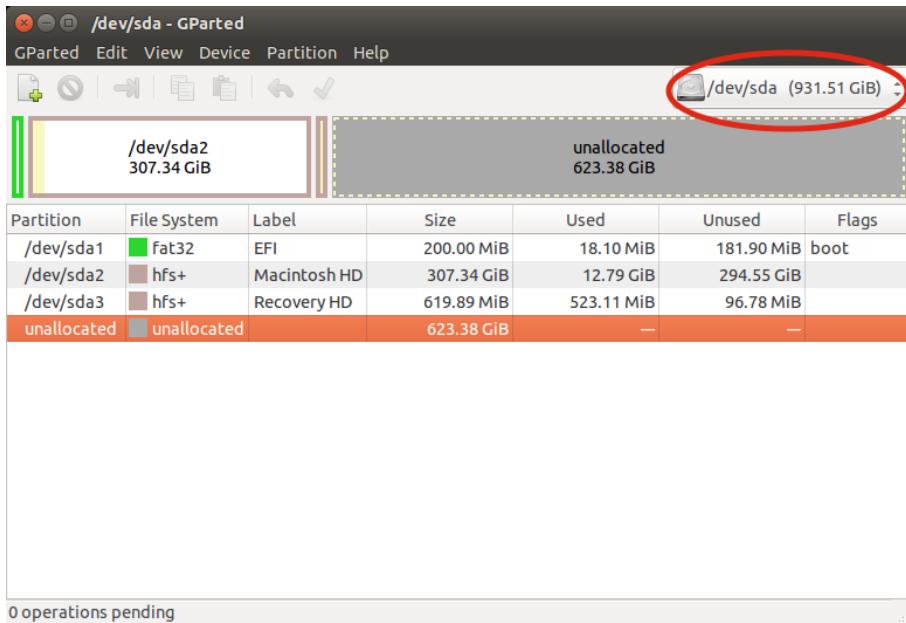
**Note** you may be met by a prompt similar to the one below, indicating that the primary partition table is corrupt, and the secondary one will be used. In that case do not worry, simply click *Ok* and continue.



Once the program has started up, and scanned the disk for a partition table, you should get a view similar to the one below.



If you do not get an image like this, or it looks significantly different, ensure the correct disk is currently in use by GParted. This can be selected in the top right hand corner of the UI.



When all set, the next step is to define the new partitions on the disk.

As you can see, I have roughly 670 GB free (The screenshot says 623 GiB - Note the difference between GiB and GB), this corresponds to the amount of space we left empty from Disk Utility in OS X.

Your amount of space may be different, however the ways in which it will be devided will be roughly the same. As always, consider how my instructions may need to be bent to fit your exact usecase, this guide is not a bible, and more often than not, will need to be veared off from.

We will need to create three different partitions

- Windows
- Ubuntu
- Swap

Let's go over the partitions in order, and discuss the requirements for them, and the specifics I will select.

## Windows

This partition will be containing the entirety of the Windows filesystem. That means Operating System, any backup / recovery systems, and all user data.

For any newer version of Windows (From Windows NT 3.1 and forward), the partition containing the filesystem will be of the type [NTFS](#).

I will be making this partition 330 GB large, leaving me 340 for Ubuntu and Swap, however you may choose any size larger than 50 GB or so, depending on OS Version.

The reason for this is that most Windows versions have been in the range of 20 - 40 GB large, for the OS itself, and you want to leave some space for data.

## **Ubuntu**

This partition will have much the same job for Ubuntu, as the Windows partition will have for Windows.

It will contain the operating system, the user data, and a few other bits and bobs.

The one thing however it will not contain, that both OS X and Windows automatically include in their partitions, is swap space, also known as a page file in Windows.

Linux uses its own filesystem type for the operating system. It's named *ext4*.

I will make my Ubuntu partition 330 GB as well, leaving me 10 for swap.

## **Swap**

The swap area is the area any linux operating system, in our case Ubuntu, will use as temporary memory, if the entirety of the RAM is filled. It ensures that the computer doesn't simply crash, if processes attempt to use more memory than available. Instead the data not currently in active use, will be written to this swap partition of the disk.

Both OS X and Windows include similar functionality, they however do not require a separate partition for this, and manage it internally.

For this partition we will need the filesystem type aptly named *swap*, and let it use the rest of the disk space, approximately 10 GB.

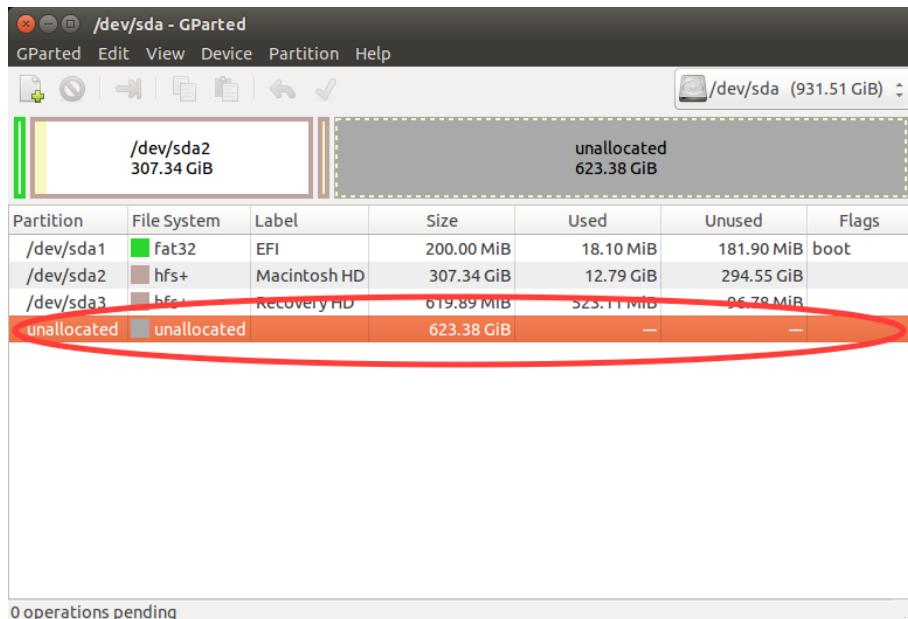
There's much discussion regarding how large a swap partition should be, and you're free to explore online for the correct

amount for your particular usecase. However as roughly 10 GB happens to just be what 3x330 left me with, that's what I'll use.

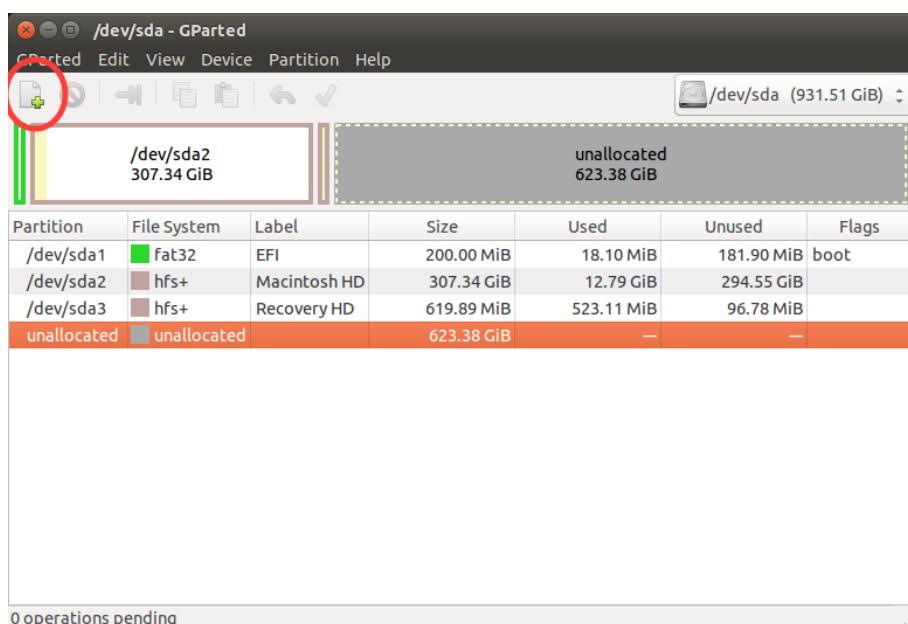
With that out of the way, we're ready to start creating the partitions.

For each new partition we want to create, we'll want to ensure the unallocated space is selected, then click the *Create a new partition in the selected unallocatd space*.

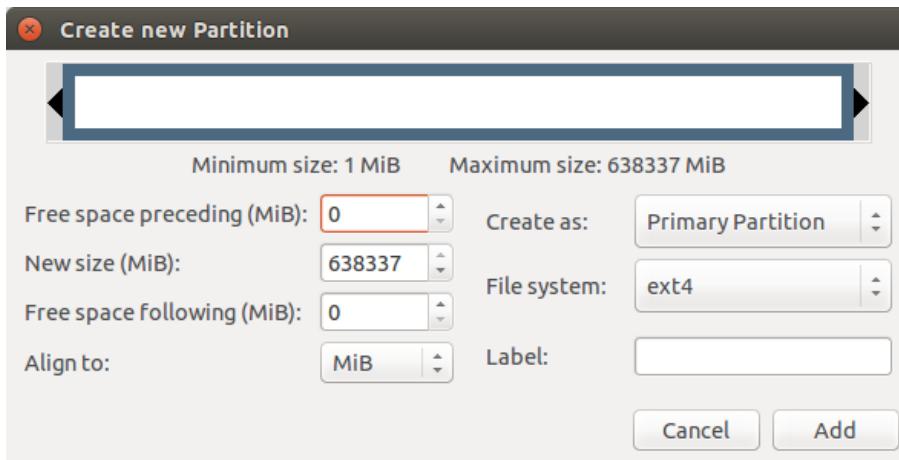
To select the unallocated space, either click on it in the bar display, or in the partition list.



Then click the button to create a new partition, in the upper left of the UI.



That'll bring you to the UI for creating a new partition.



Here you have a number of options, for defining the partition:

- Free space preceding (The amount of space to leave behind the previous partition, before this one starts)
- New size (The size of the partition we are creating)
- Free space following (The amount of empty space to leave after the partition)

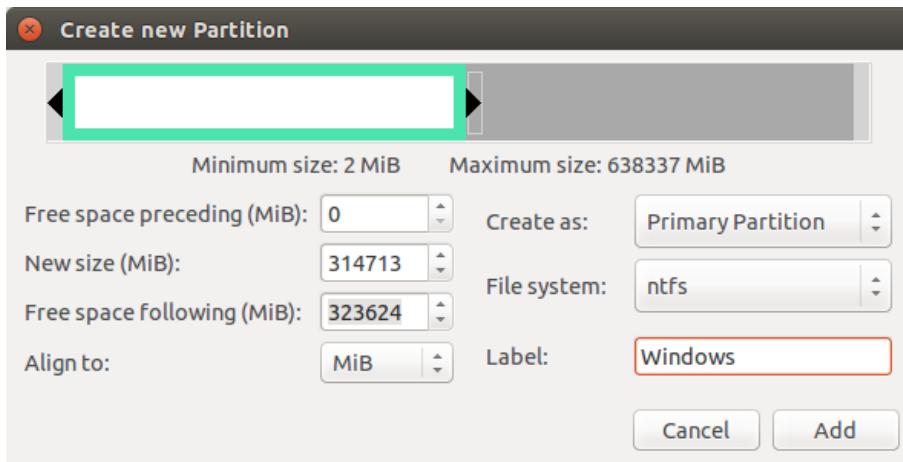
Any two of the three settings above, will automatically calculate the third.

- Align to (Whether to resize the partition to make sure it fits exactly within the bounds of one MiB, one cylinder, or not to do so at all)
- Create as (The partition type to use)
- File system (The file system type to use)
- Label (Text label for the partition)

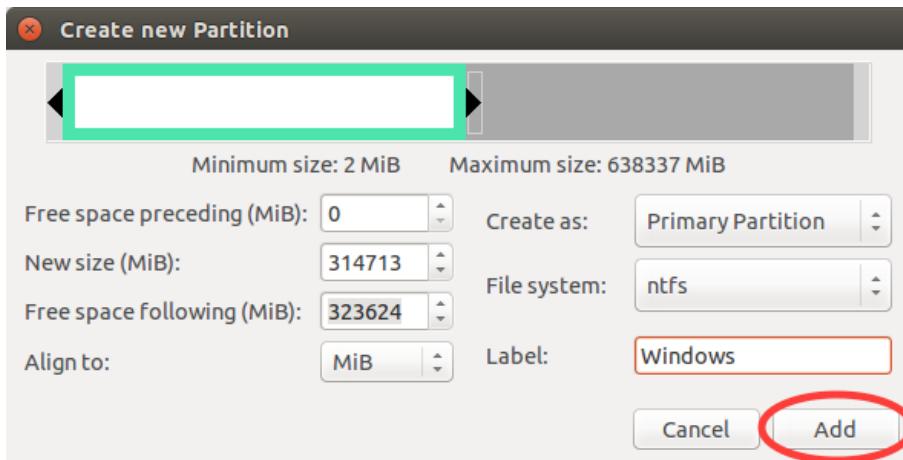
You can find more documentation on what the particular terms mean online. The following will describe which settings I believe that you should use, and which you may select yourself.

## Partition 1 - Windows

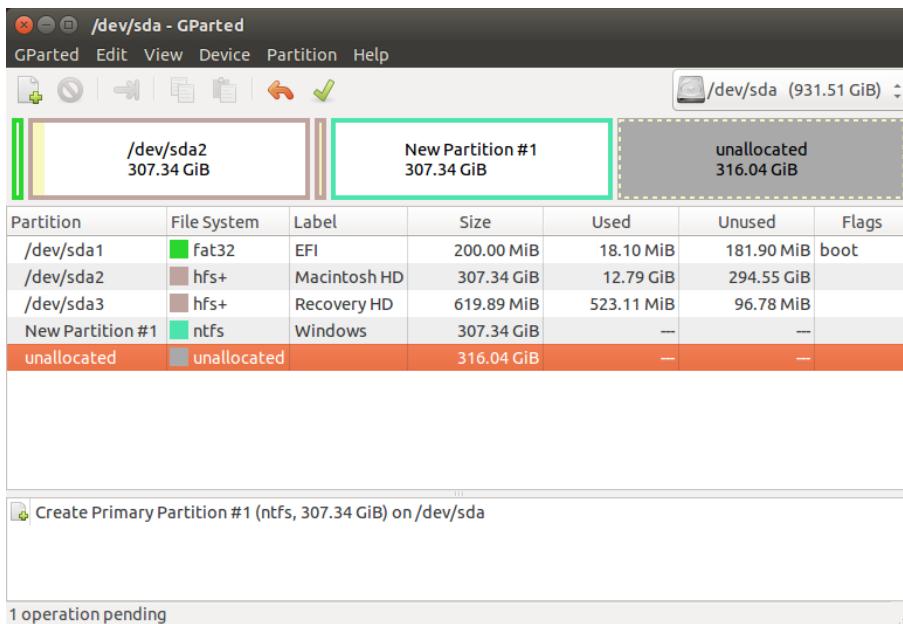
- Preceding space: 0
- Size: 314713 MiB (Which is the same as 330 GB) - Select this as appropriate for you
- Following: Leave this as automatically calculated
- Align to: MiB
- Create as: Primary partition
- File system: ntfs
- Label: Windows - Select as you wish



Then click Add



You should return to a GParted UI similar to the one below

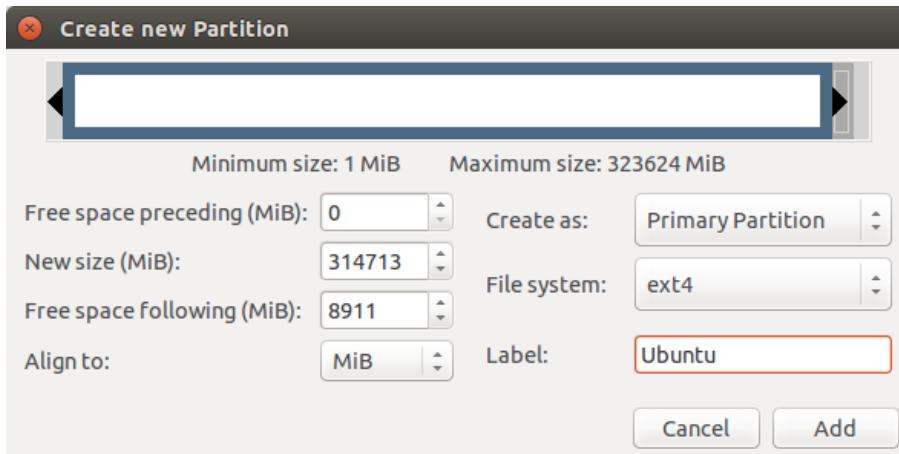


Now we repeat for Ubuntu

## Partition 2 - Ubuntu

- Preceding space: 0

- Size: 314713 MiB (Which is the same as 330 GB) - Select this as appropriate for you
- Following: Leave this as automatically calculated
- Align to: MiB
- Create as: Primary partition
- File system: ext4
- Label: Ubuntu - Select as you wish

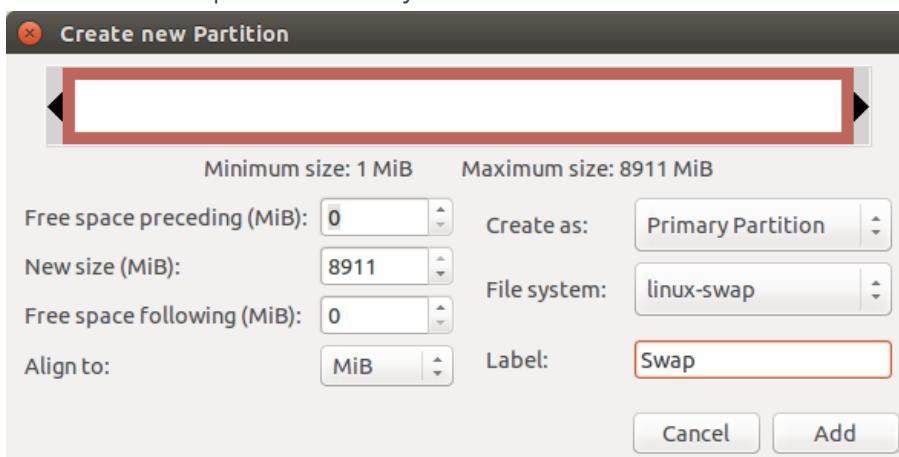


Again click *Add*

Finally the swap partition

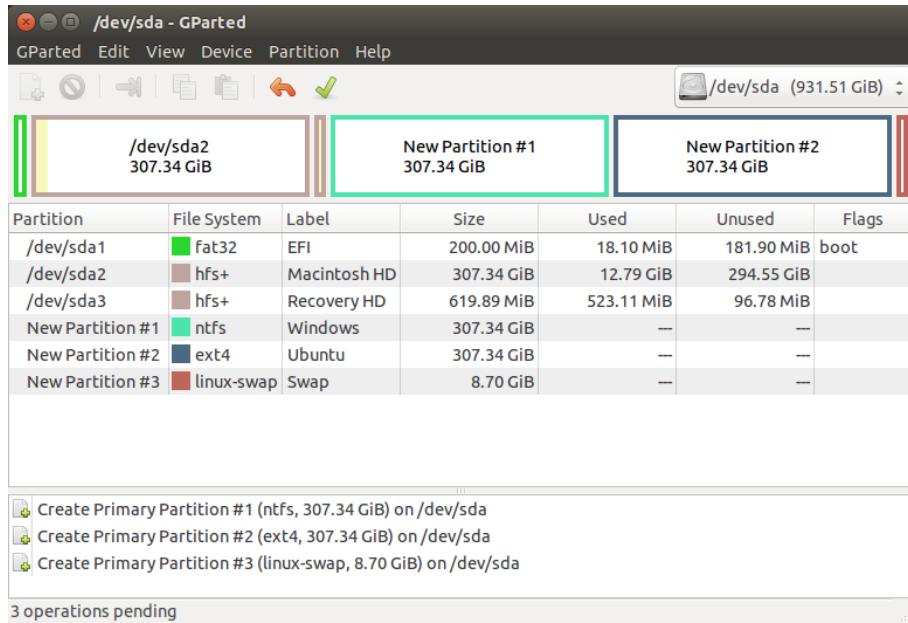
### Partition 3 - Swap

- Preceding space: 0
- Size: 8911 MiB (The rest of the space left) - Select this as appropriate for you
- Following: 0
- Align to: MiB
- Create as: Primary partition
- File system: linux-swap
- Label: Swap - Select as you wish



And click *Add* one last time.

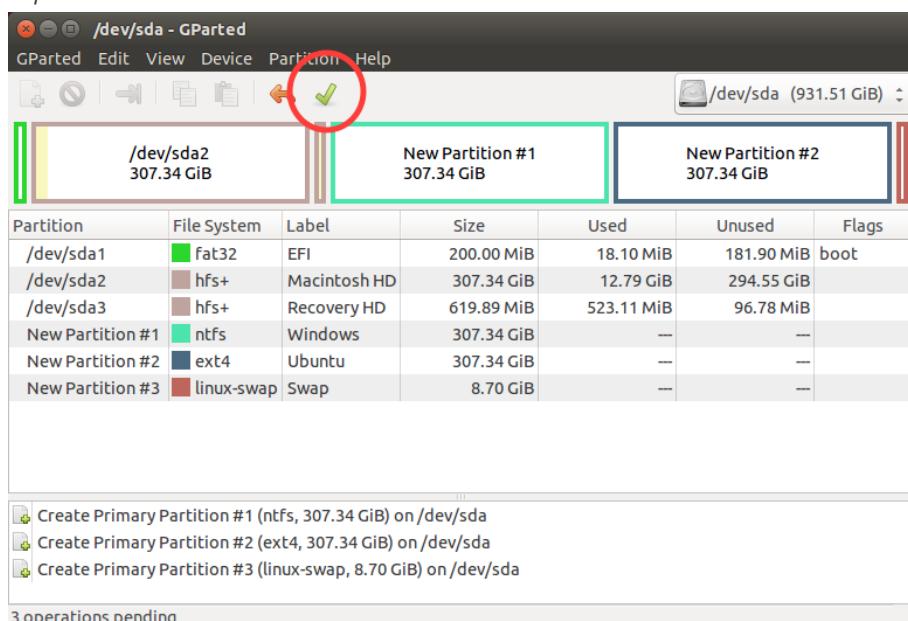
Once you've gone through these steps, you should have a display similar to the one below, showing the three new partitions taking up the space, which was previously unused.



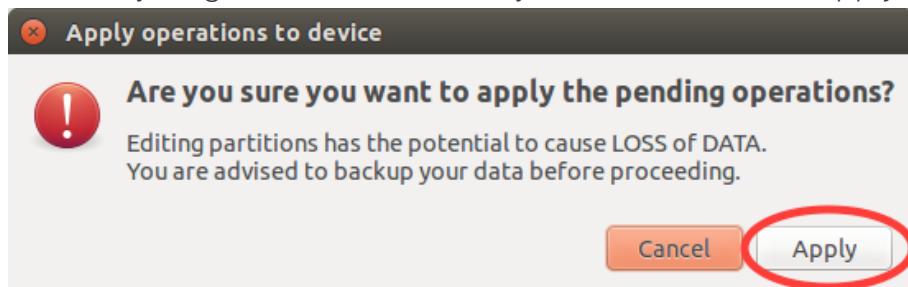
It's worth noting that none of these partitions have been persisted to the disk yet, if you were to close GParted at this point, nothing would have changed. This is good in case you fear that you created a partition with the wrong configuration. Simply clear the changes and try again.

You can also see GParted informs you that there are 3 *operations pending* in the bottom part of the display, and what exactly those are.

If you are happy with the changes, it's time to click *Apply All Operations*



You might be asked if you are really sure in what you're doing. Of course this may depend on the state of life, the universe, and everything. However I assume you will want to click *Apply*

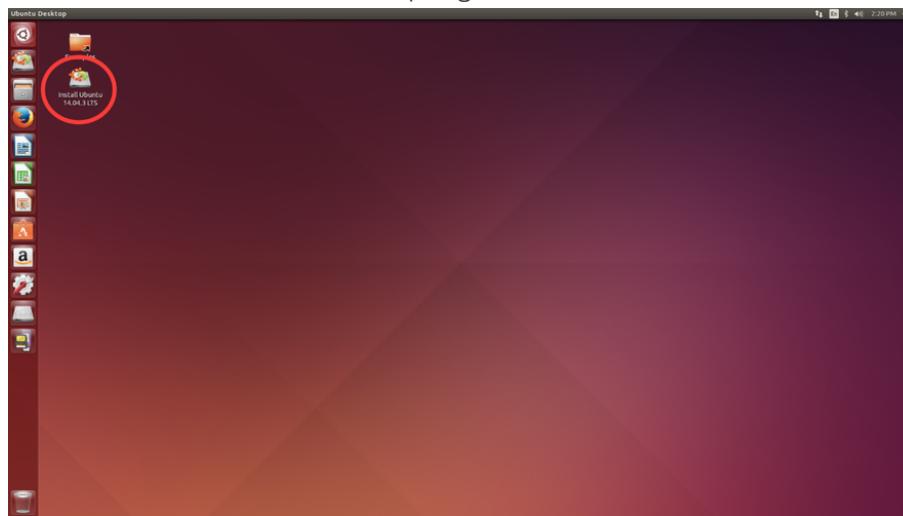


The process will take a few moments, and GParted will display what it's in the process of doing. Once it's done, the partitions are ready for operating systems to be installed on them.

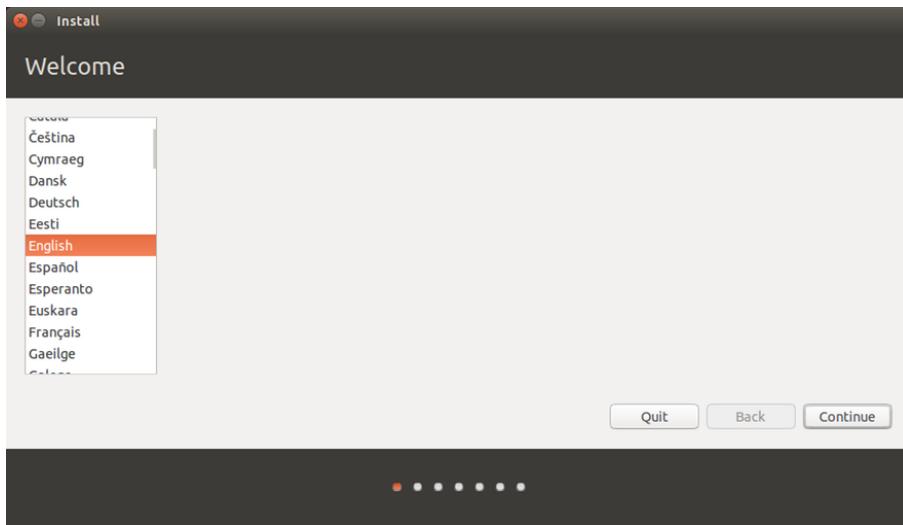
## Installing Ubuntu

If you have shut down the computer since creating the partitions, boot it into the Ubuntu USB again, and select *Try Ubuntu without installing* as before.

Once you're in Ubuntu again, or if you're still running the live version of Ubuntu. Start the program called *Install Ubuntu*.



The first screen you will be met with, will prompt you to select language. Choose whichever one you are comfortable with, I will be using English, and describing the English options throughout the guide.

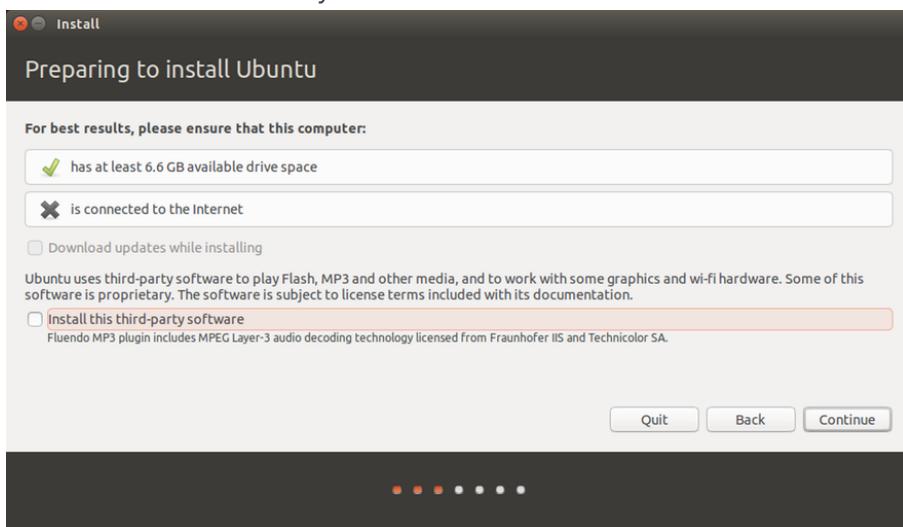


Then Ubuntu will check that you have enough disk space, and whether you have an active internet connection.

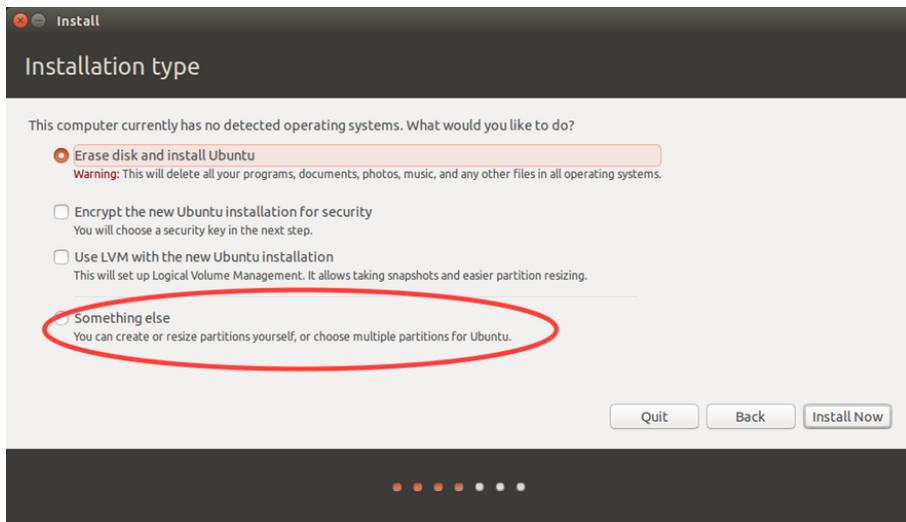
If you have enough space, you will be permitted to continue the installation. If you also have an active internet connection, you will be asked if you wish to install updates during the installation.

You may also select if you wish to install 3rd party software during the install. This can be done later as well, so don't worry if you don't choose to do so now.

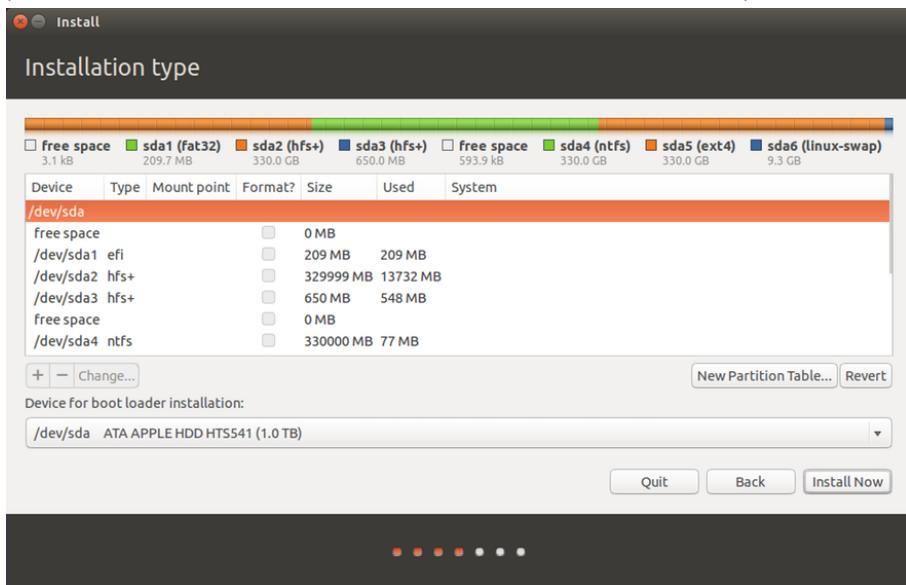
I don't have network connected at the moment, and I will be updating the OS later anyway, so I'm opting not to update. You are free to choose as you wish.



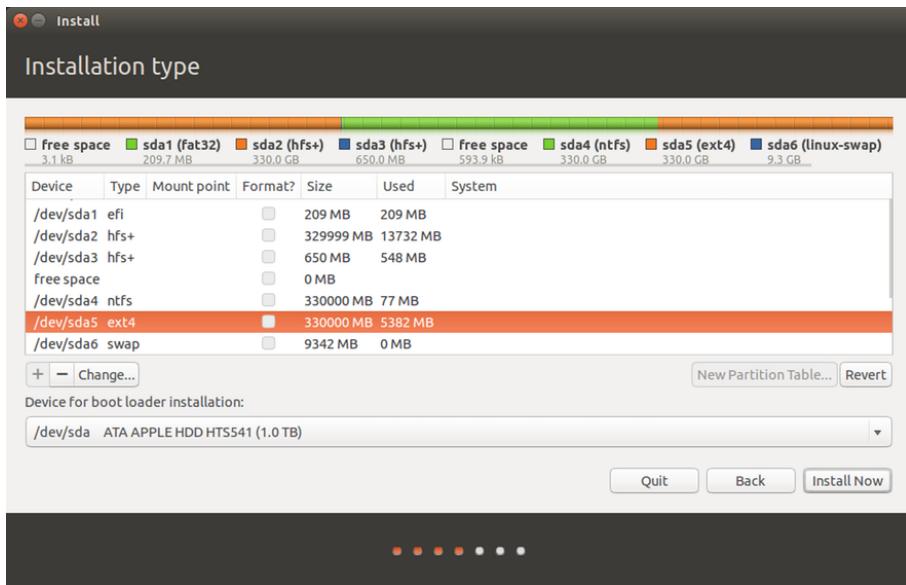
Next up you need to select how the disk will be handled. By default Ubuntu can make sure to delete the whole disk, and install itself for you. However we've just spent time setting up the partitions as we want, so we want to manage the disk partitions ourselves. Select *Something else* to be able to select which partition to use ourselves.



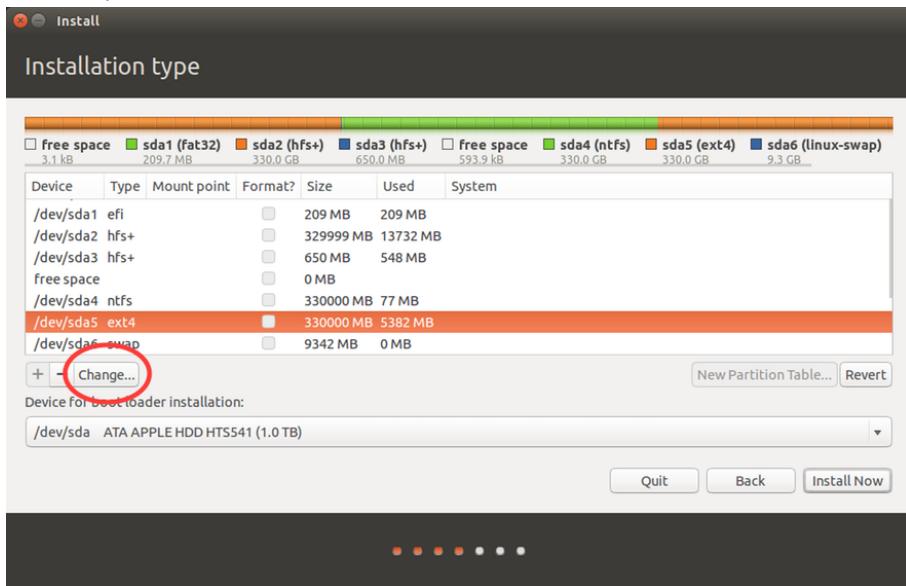
Then we'll be met with a more advanced view, of the different partitions on the disk. Here we'll be telling Ubuntu which partition to install itself on, and which to use for swap.



Navigate down the list to the partition you will want to install Ubuntu on. It can be identified by the filesystem, which will be ext4.

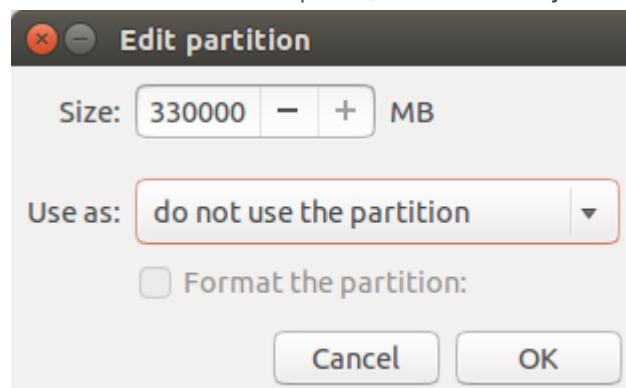


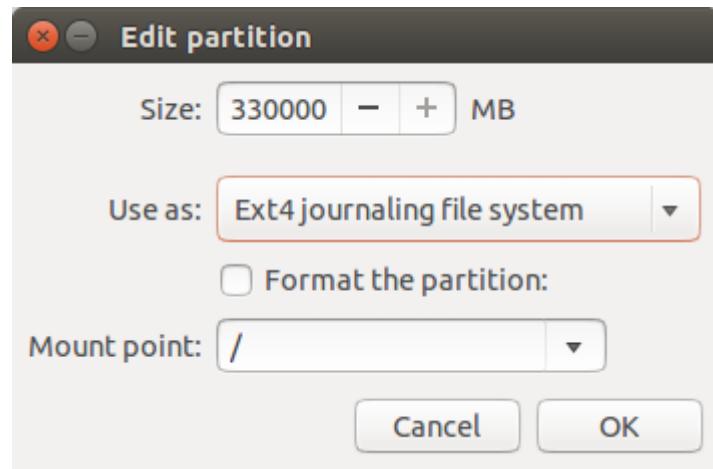
We'll then click the *Change...* button, to change the fact that the partition will not be in use, to the partition being the primary mount point for Ubuntu.



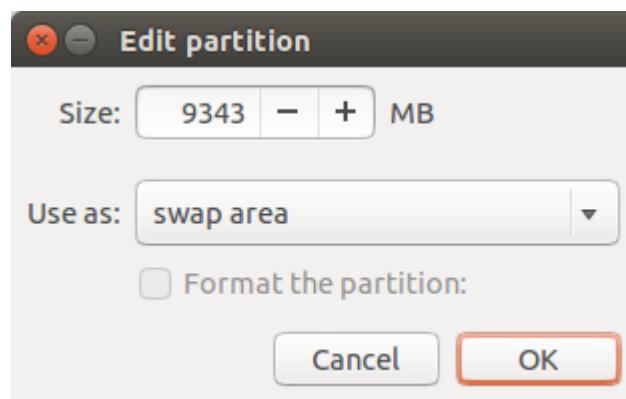
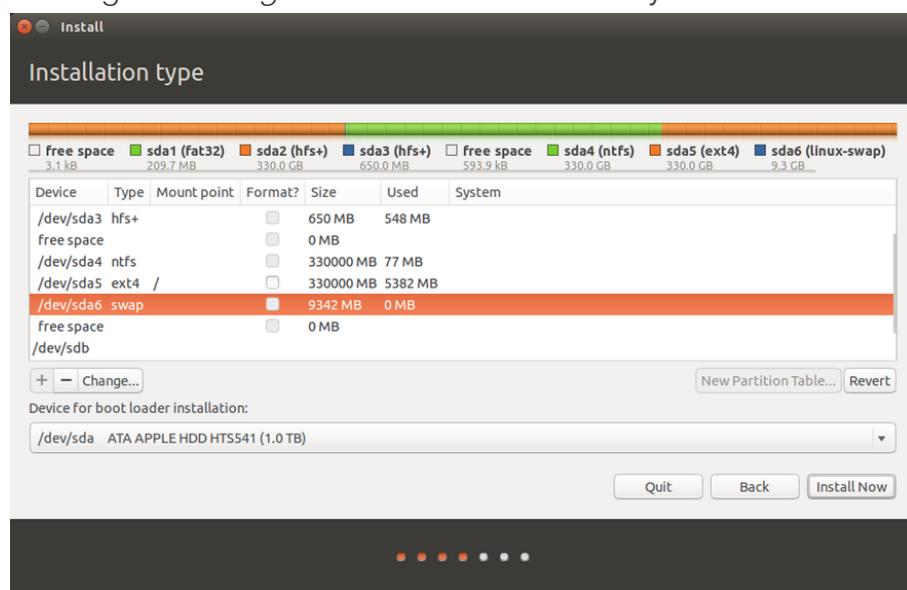
The dialogue will indicate that the partition will not be in use by Ubuntu at all.

Change this to indicate that the partition is an ext4 file system, and to make it the root mount point, indicated by /

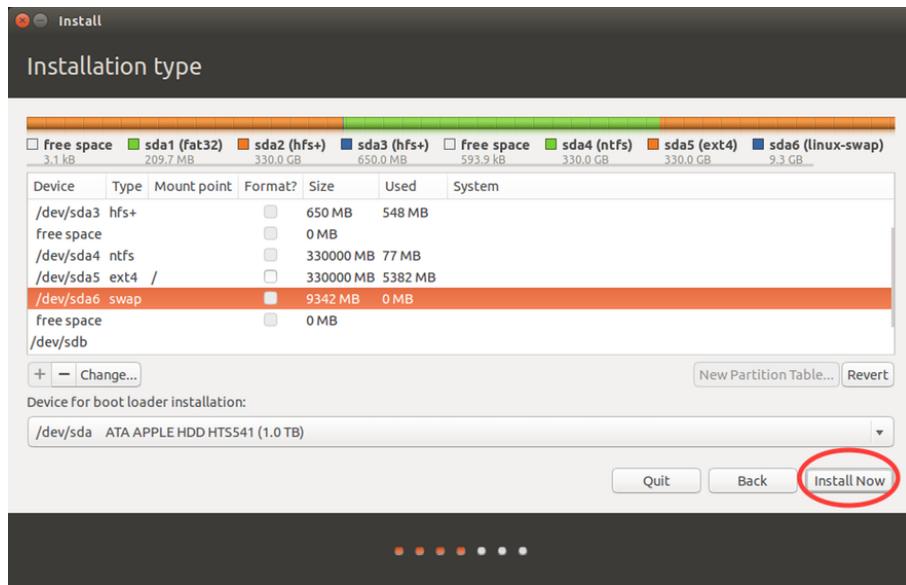




The swap area should be selected correctly on its own. However you can select it in the partition list, and check by clicking the *Change...* button in the same way.

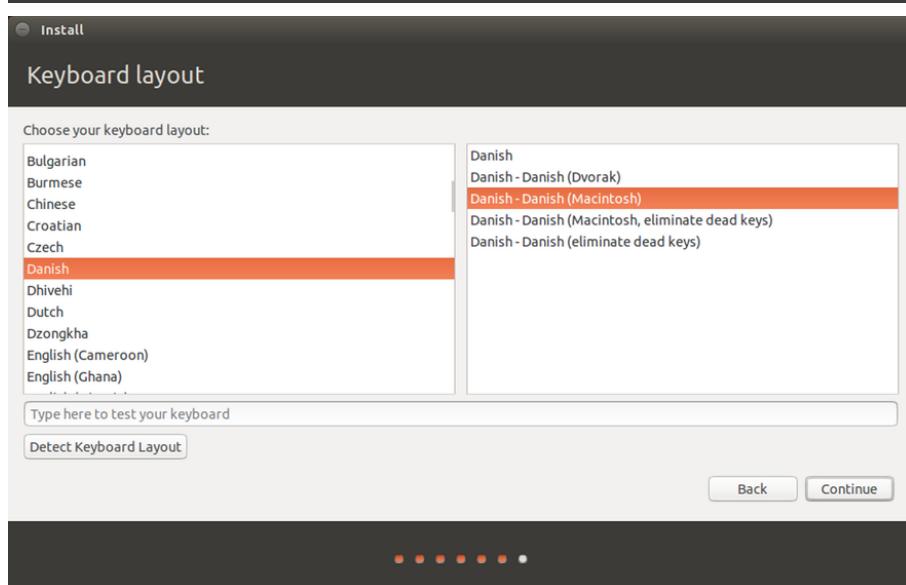
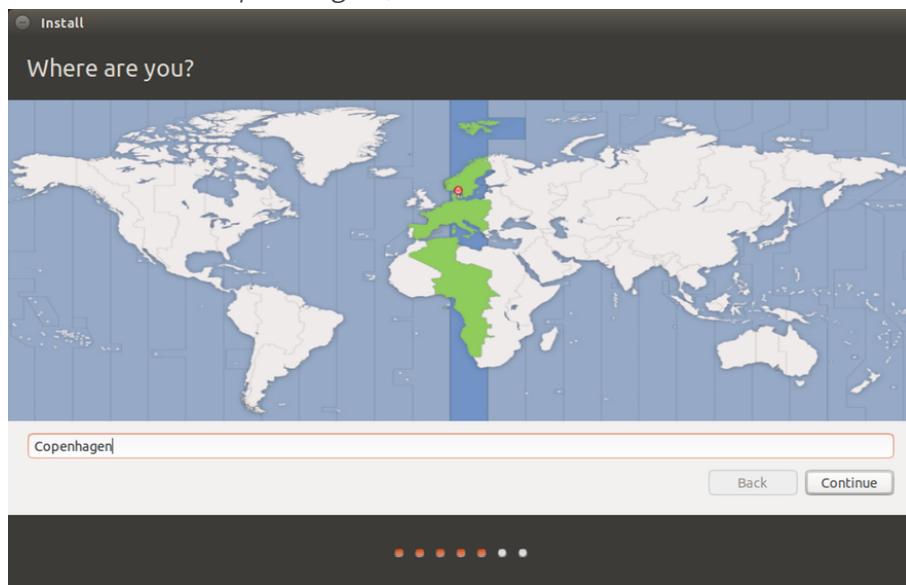


Once you are satisfied, you can continue the installation.



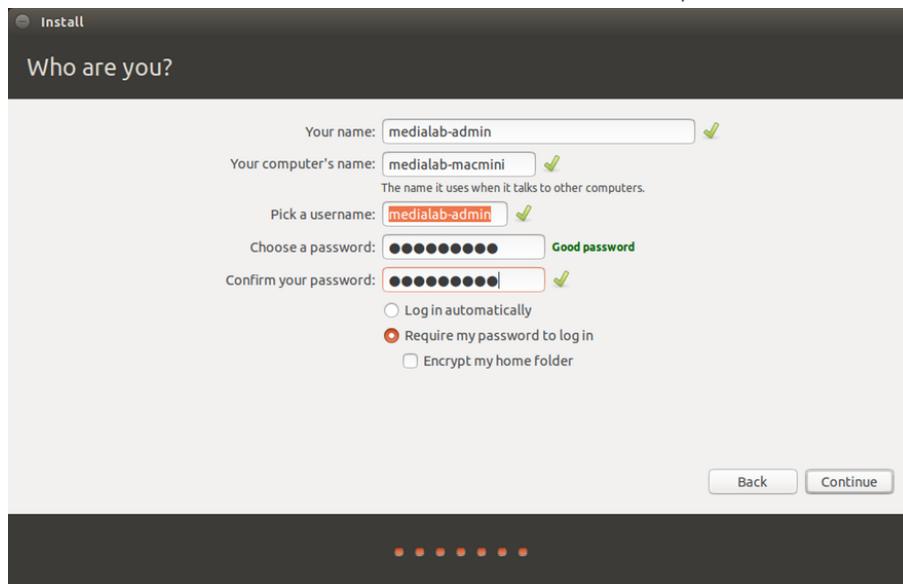
You'll also be asked to select the location the computer is in, and the keyboard layout to use.

Here I select *Copenhagen*, and *Danish*.



Finally create the local user account to be used on the computer for administrative purposes.

I will call mine *medialab-admin*, and set a safe password.

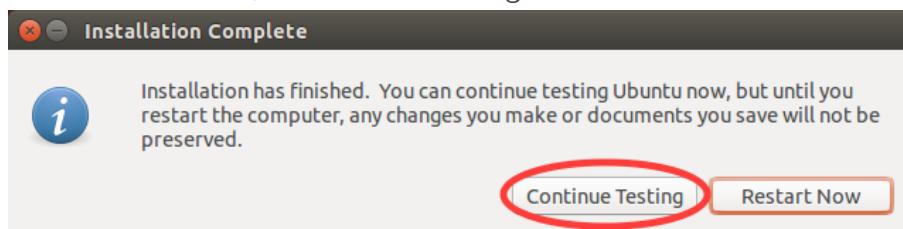


Now you can go get a cup of your favourite beverage, this will probably take a few minutes.

Once it's over, if you started the installation from a live version of Ubuntu, you will be asked if you want to restart to boot into the newly installed OS, or want to continue testing the live version.

**Note** that we want to **continue testing** at this point.

Before we restart, there's something we need to fix first.



## Revert hybrid MBR to protective MBR

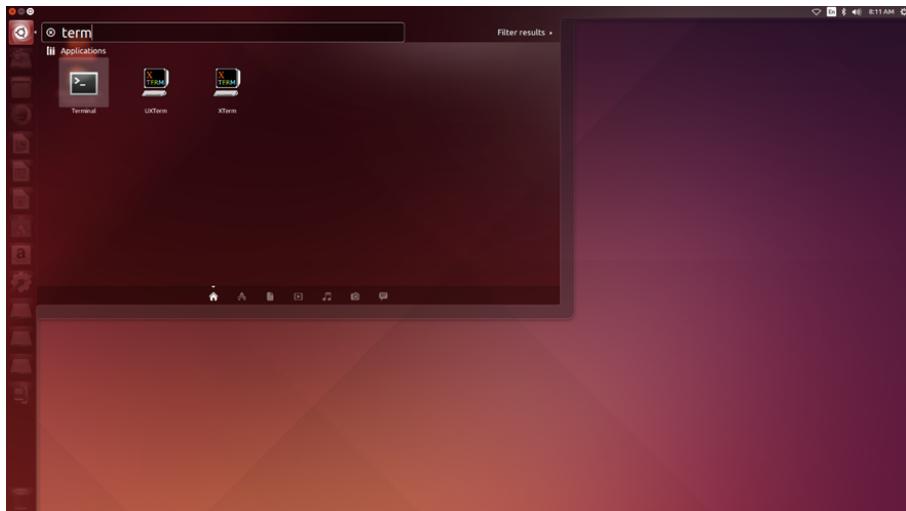
These steps are based by a guide by [Rod Smith](#), for his exact words see <http://www.rodsbooks.com/ubuntu-efi/index.html#fixing>

As an attempt in being helpful, Ubuntu will during the installation convert the GPT partition table, to a hybrid MBR partition table. Some older OS only know how to work with MBR, and as such, if they are to work with other OS which were installed using GTP, the partition table needs to be made into a hybrid MBR.

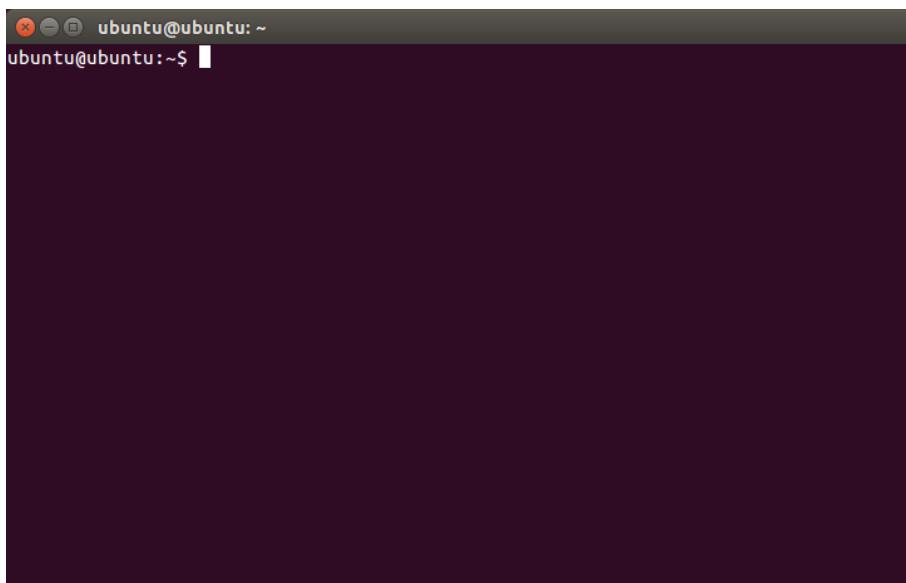
This sounds good in practice, but in reality hybrid MBR is unstable, unreliable, and if possible, it should be avoided. More concrete information on partition tables, MBR, and GPT can be found online, I won't go into the technical details.

As all the OS's we are using in this setup are new enough to have support for GPT natively, we have no need for a hybrid MBR, but we cannot instruct Ubuntu to not make one, we have to revert the change manually after the fact.

First up we'll need to start up the *Terminal*, just like when starting GParted, open the menu, and search for the program.



Once the terminal is started, you should have a view like the one below.

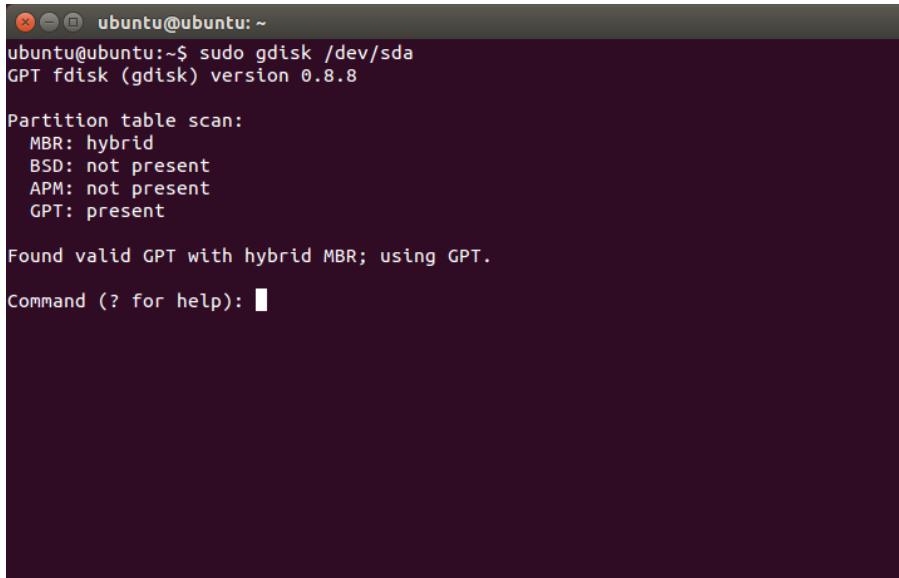


To change the partition table back to GPT with protective MBR (This basically means that the MBR is disabled), we'll be typing a few commands into the terminal.

```
sudo gdisk /dev/sda
```

**Note** here that /dev/sda is the name of the disk we're installing the OS on, this may be different for you. A good way to check if you're in doubt, is to go back to GParted and check in the top right hand corner, which disk the partitions are on.

If you used the right disk, you should see something like this, indicating a GPT partition table present, and the MBR in hybrid mode.



```
ubuntu@ubuntu:~$ sudo gdisk /dev/sda
GPT fdisk (gdisk) version 0.8.8

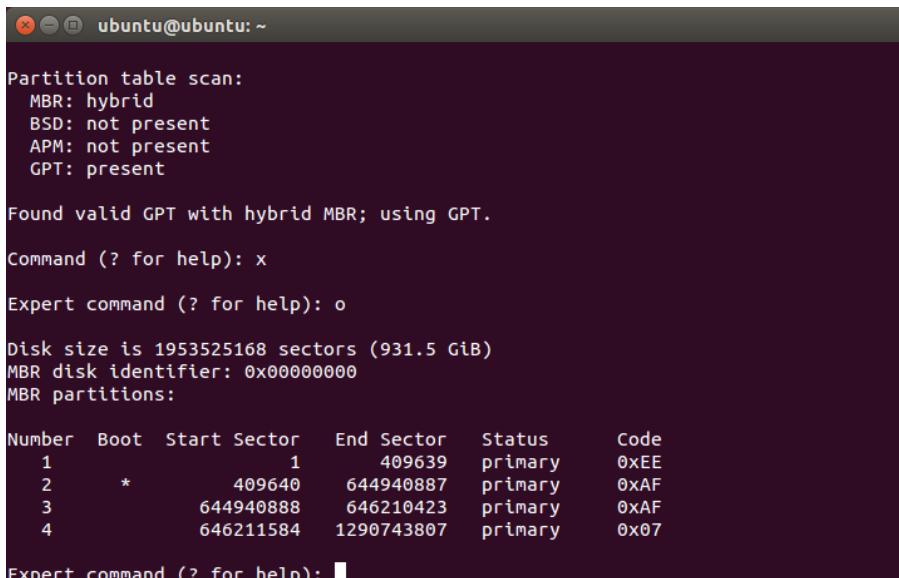
Partition table scan:
  MBR: hybrid
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with hybrid MBR; using GPT.

Command (? for help):
```

Now enter *x*, for expert mode, then *o*, this will list the content of the MBR.

The hybrid MBR should contain 4 partitions, as displayed below.



```
ubuntu@ubuntu:~$ sudo gdisk /dev/sda
GPT fdisk (gdisk) version 0.8.8

Partition table scan:
  MBR: hybrid
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with hybrid MBR; using GPT.

Command (? for help): x
Expert command (? for help): o

Disk size is 1953525168 sectors (931.5 GiB)
MBR disk identifier: 0x00000000
MBR partitions:

Number  Boot  Start Sector    End Sector    Status     Code
      1            1        409639  primary    0xEE
      2      *        409640    644940887  primary    0xAF
      3            644940888   646210423  primary    0xAF
      4            646211584  1290743807  primary    0x07

Expert command (? for help):
```

Enter *n* in the prompt, this should make a new MBR for you, however it won't show any changes on its own. Type *o* again to see the new MBR partitions.

If you get a display like this, it's been done correctly.

```
ubuntu@ubuntu: ~
Expert command (? for help): o

Disk size is 1953525168 sectors (931.5 GiB)
MBR disk identifier: 0x00000000
MBR partitions:

Number  Boot  Start Sector    End Sector    Status      Code
      1            1        409639  primary      0xEE
      2      *        409640    644940887 primary      0xAF
      3            644940888   646210423 primary      0xAF
      4            646211584   1290743807 primary      0x07

Expert command (? for help): n

Expert command (? for help): o

Disk size is 1953525168 sectors (931.5 GiB)
MBR disk identifier: 0x00000000
MBR partitions:

Number  Boot  Start Sector    End Sector    Status      Code
      1            1        1953525167 primary      0xEE

Expert command (? for help):
```

Finally, enter *w* for write, writing your changes to the disk. You'll be asked if you're entirely sure about your changes, as they'll overwrite your current disk. Go ahead.

```
ubuntu@ubuntu: ~
Number  Boot  Start Sector    End Sector    Status      Code
      1            1        409639  primary      0xEE
      2      *        409640    644940887 primary      0xAF
      3            644940888   646210423 primary      0xAF
      4            646211584   1290743807 primary      0x07

Expert command (? for help): n

Expert command (? for help): o

Disk size is 1953525168 sectors (931.5 GiB)
MBR disk identifier: 0x00000000
MBR partitions:

Number  Boot  Start Sector    End Sector    Status      Code
      1            1        1953525167 primary      0xEE

Expert command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N):
```

If everything worked out successfully, your terminal should look like this in the end.

```
ubuntu@ubuntu: ~
      4          646211584  1290743807  primary      0x07

Expert command (? for help): n

Expert command (? for help): o

Disk size is 1953525168 sectors (931.5 GiB)
MBR disk identifier: 0x00000000
MBR partitions:

Number  Boot Start Sector    End Sector    Status     Code
      1           1   1953525167  primary      0xEE

Expert command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): Y
OK; writing new GUID partition table (GPT) to /dev/sda.
Warning: The kernel is still using the old partition table.
The new table will be used at the next reboot.
The operation has completed successfully.
ubuntu@ubuntu:~$
```

**Now** you're set to reboot the computer out of the Ubuntu installer.

Do however note that you will not yet be able to boot into Ubuntu, even if you hold down the option key on boot, and try to select the partition. The default mac bootloader cannot read the Ubuntu partition, and boot into it.

We will be addressing this issue shortly.

## Creating a Windows boot media

Similarly as with Ubuntu, we will need to create a boot media for Windows to be able to install it. You could overwrite the same DVD or USB drive, or make a new one. I highly recommend making a secondary boot media, if you have the resources available, especially if you are going to be installing on multiple computers. Otherwise you will have to overwrite the same one boot media back and forth.

For this step we are also going to require a computer already running Windows.

There may be ways to do this on other systems, however the best course of action I could find was the following.

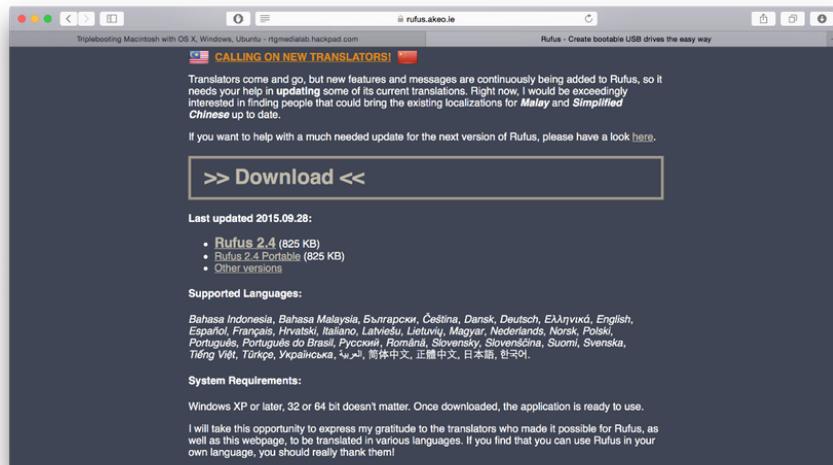
Windows versions newer than 8 support GPT partition tables, however the default tool from Microsoft for installing Windows on a DVD or USB drive, installs it as a MBR partition table, not giving you a choice.

For that reason, we use a tool called *rufus* to install Windows onto the USB media. If you are using a DVD, the steps will be

very similar.

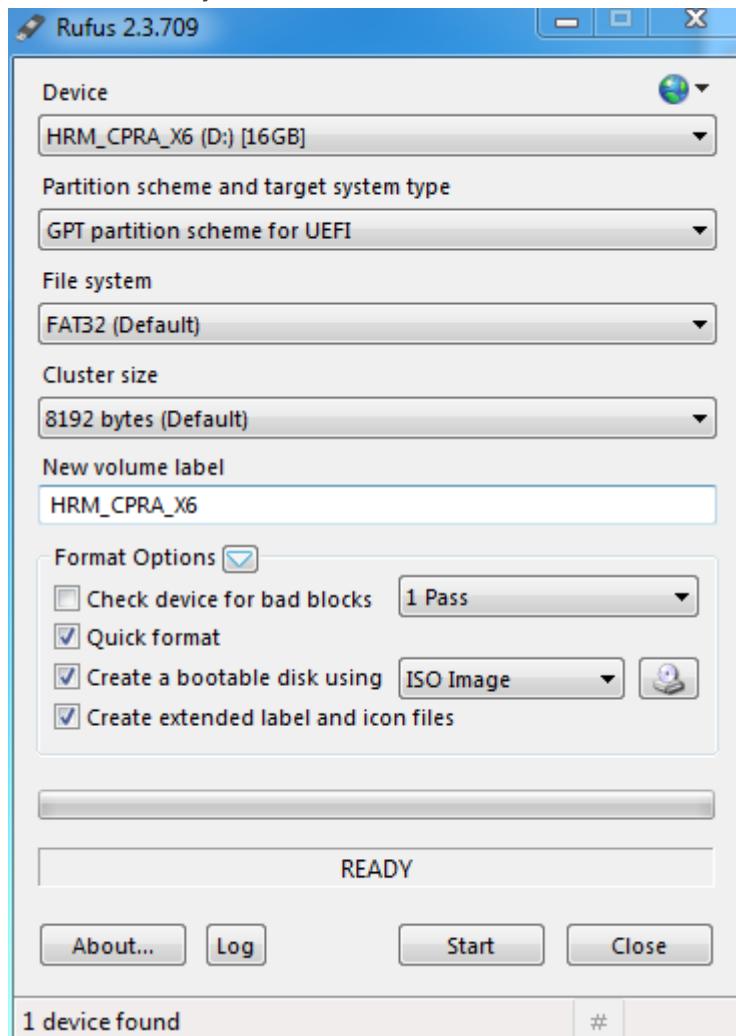
Go to <https://rufus.akeo.ie> and navigate to *Download*.

Click the link to download the current version of rufus (At the time of writing, it's 2.4



Once downloaded, run the executable, no installation shold be required.

Rufus should meet you with a UI similar to this

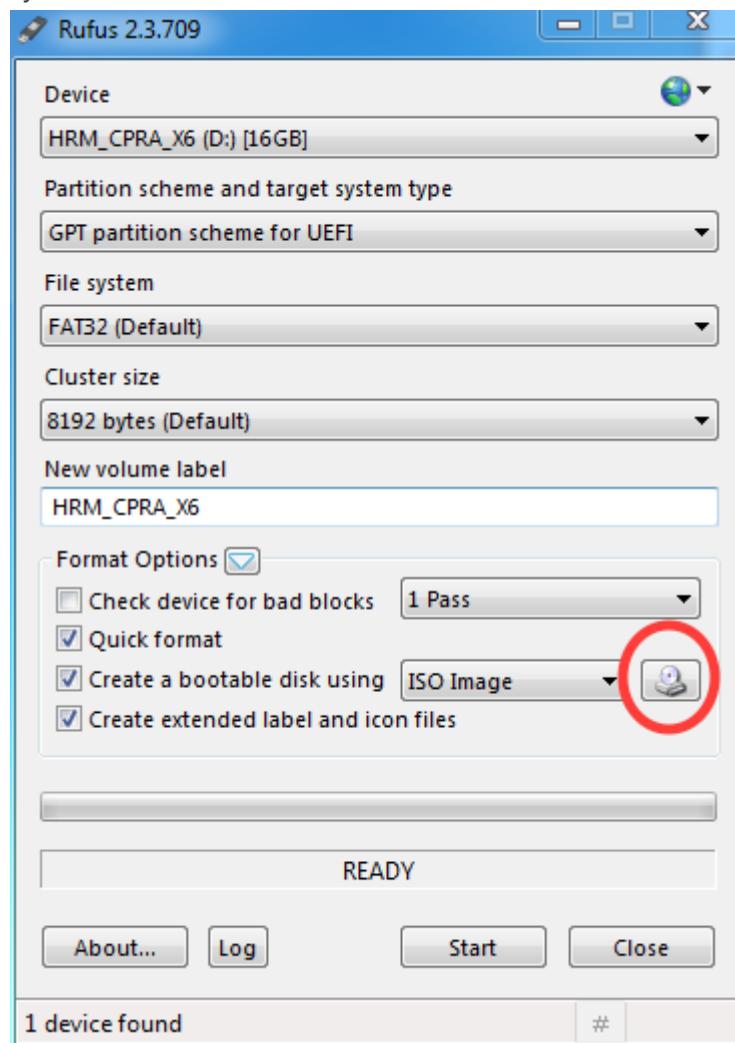


There's a number of options to choose from, however most of them you can leave default.

The options you will be concerned with are:

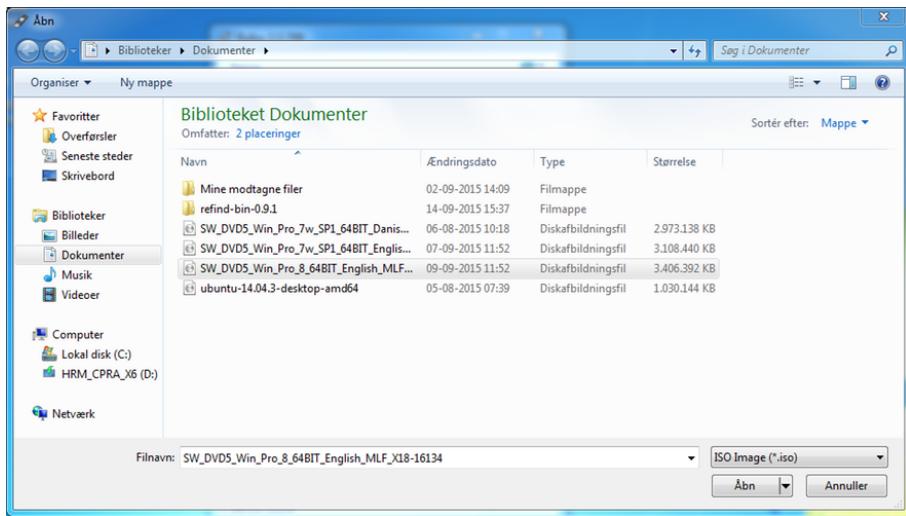
- Device (What are we installing the image on?)
- Partition scheme (We need to ensure to use GPT)
- Create a bootable disk using ISO image (Here we need to select which image we are using)

First select the device that you want to install the image on, in my case it's the USB media mounted as D, select the GPT partition scheme, and click the button highlighted below, to select your ISO file.

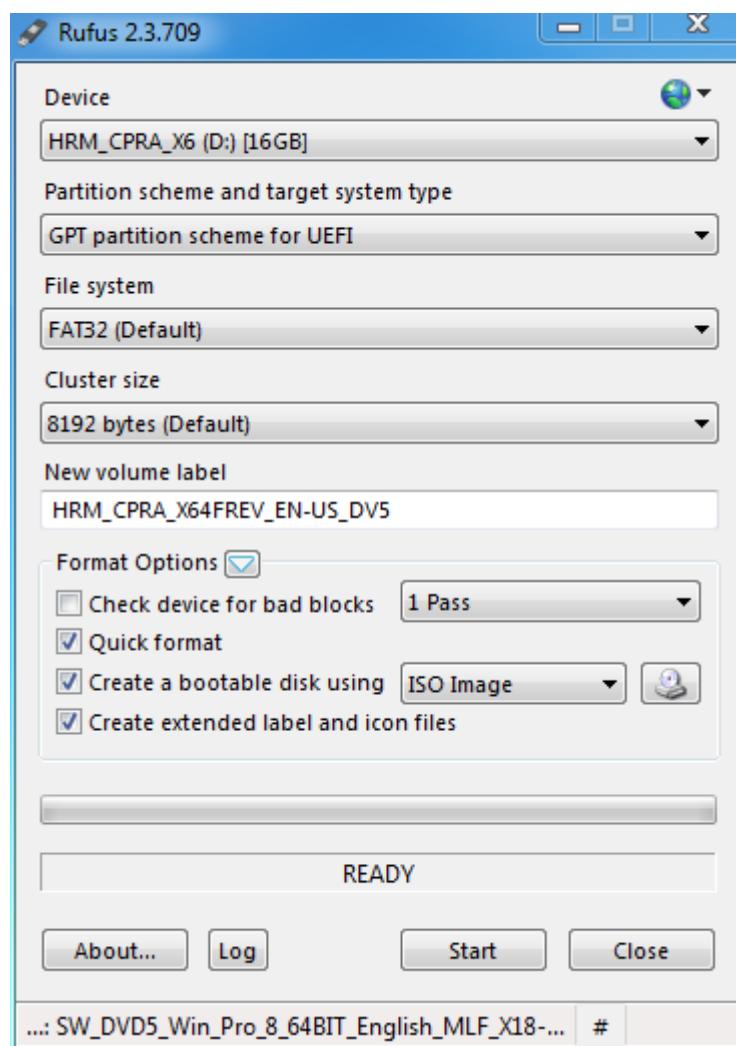


I will be using Windows 8 64 bit, English. However any version from 8 and up should work.

**Note** windows 7 may work, I have not had success, however I have not tested thoroughly.

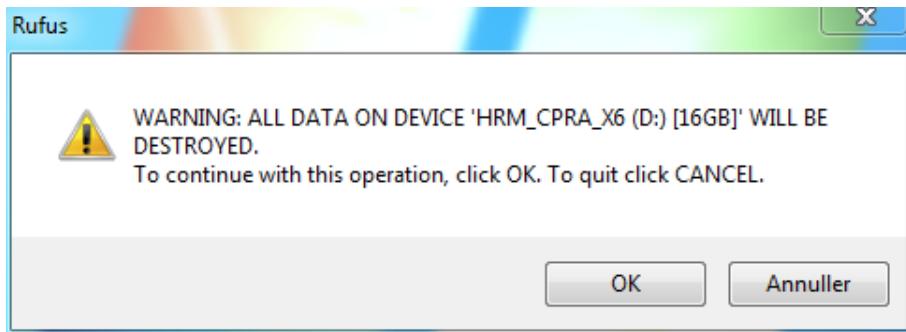


Once the file is selected, double check that all your options resemble mine.



When you're happy, click *Start*.

You may be met with a dialogue asking you to confirm that you wish to continue, as the target media will be overwriting. Click *OK* here.



Again there's a little bit of waiting, as the ISO file is writing to the media.

Once it's done, you are ready to move on to installing Windows from the media you just created, onto the Mac.

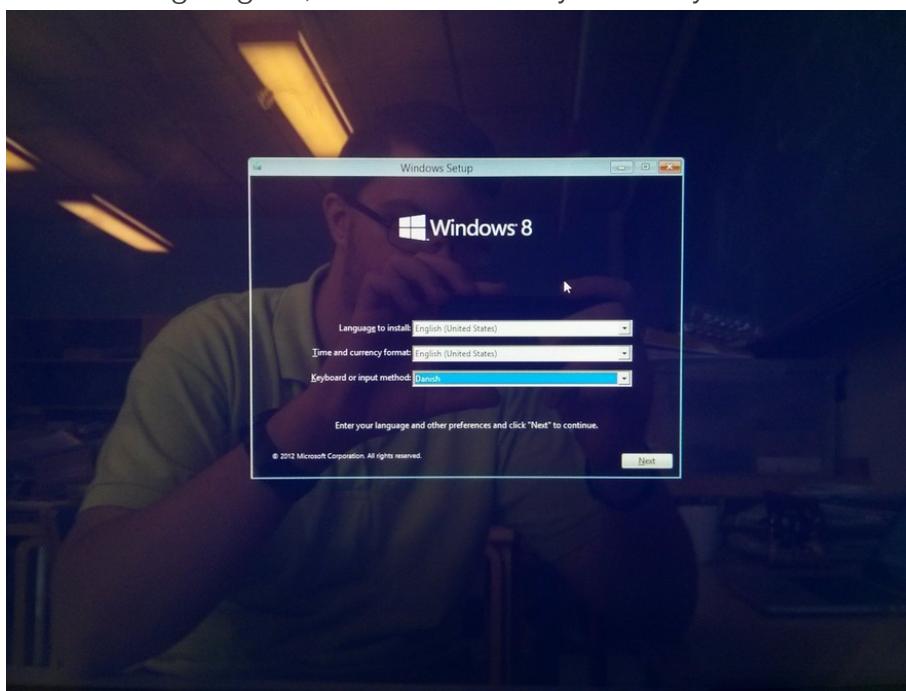
## Installing Windows

With a Windows boot media in hand, we're ready to start installing on the Mac.

Just as with Ubuntu, we insert the boot media, boot the mac, hold down the option key, and select the boot media to start from it.

Depending on your version of Windows, the options you are met with may differ, I am assuming Windows 8 in this guide.

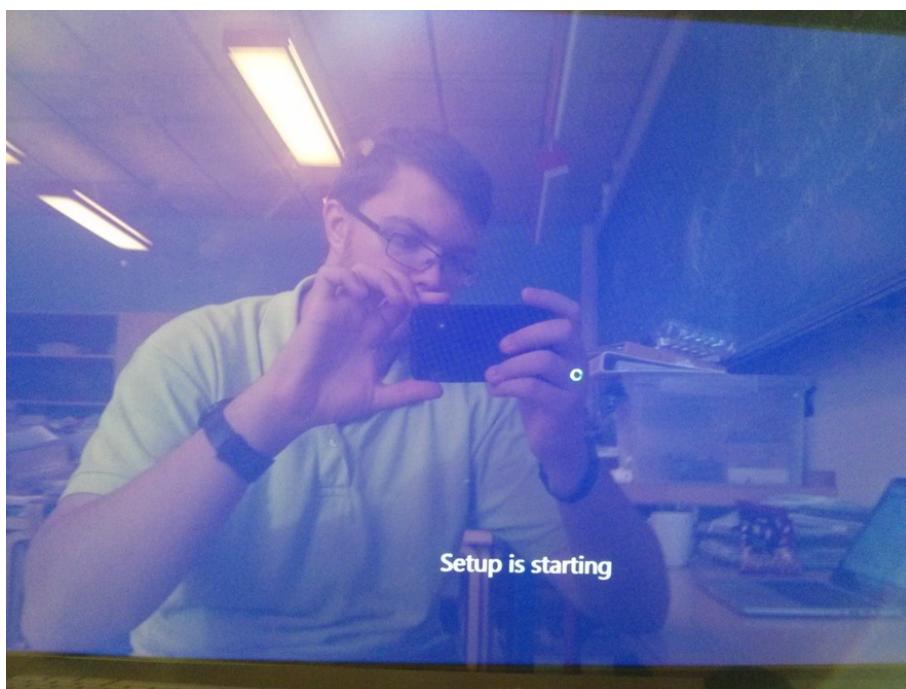
First you will be asked to provide your language preferences, I am selecting English, with a Danish keyboard layout.



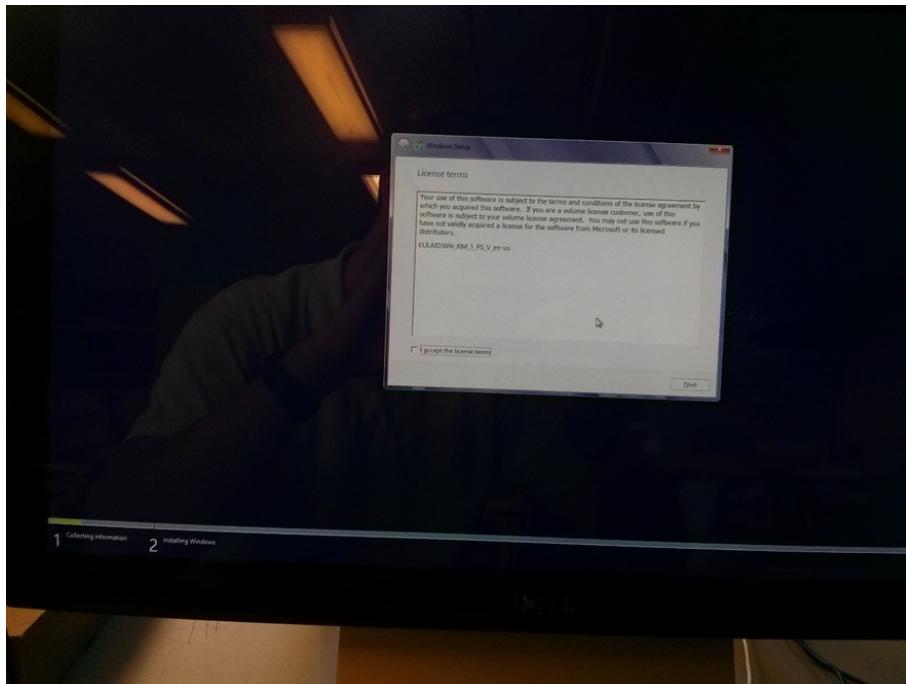
Then you will be presented with the options to install Windows, or repair an existing installation, we want to install from new.



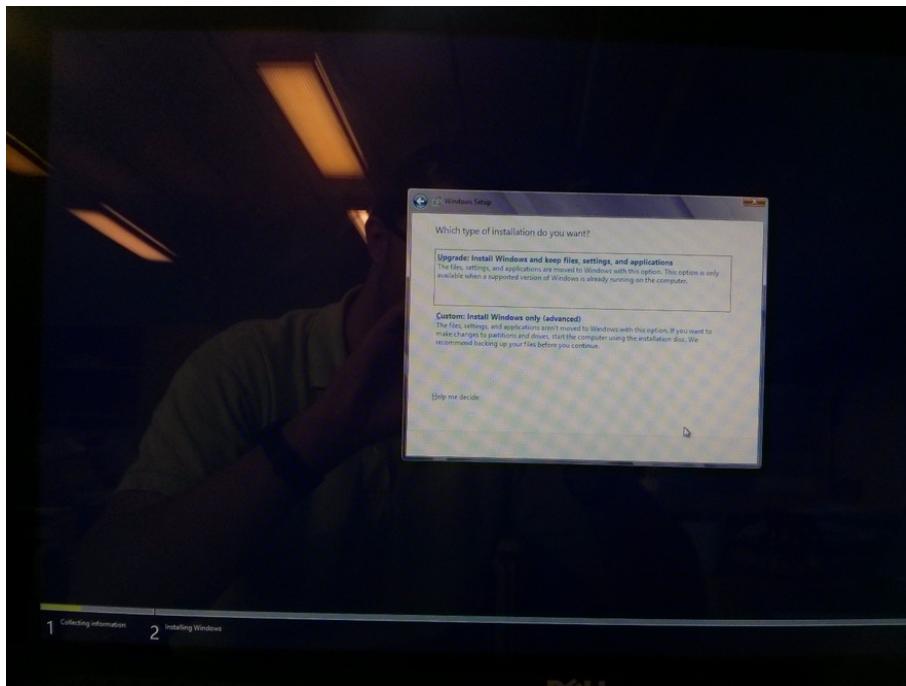
You may have to wait for a moment for the setup wizzard to continue.



Then you will need to confirm the license agreement, unless you are having second thoughts at this point (I wouldn't blame you!) go ahead and accept.



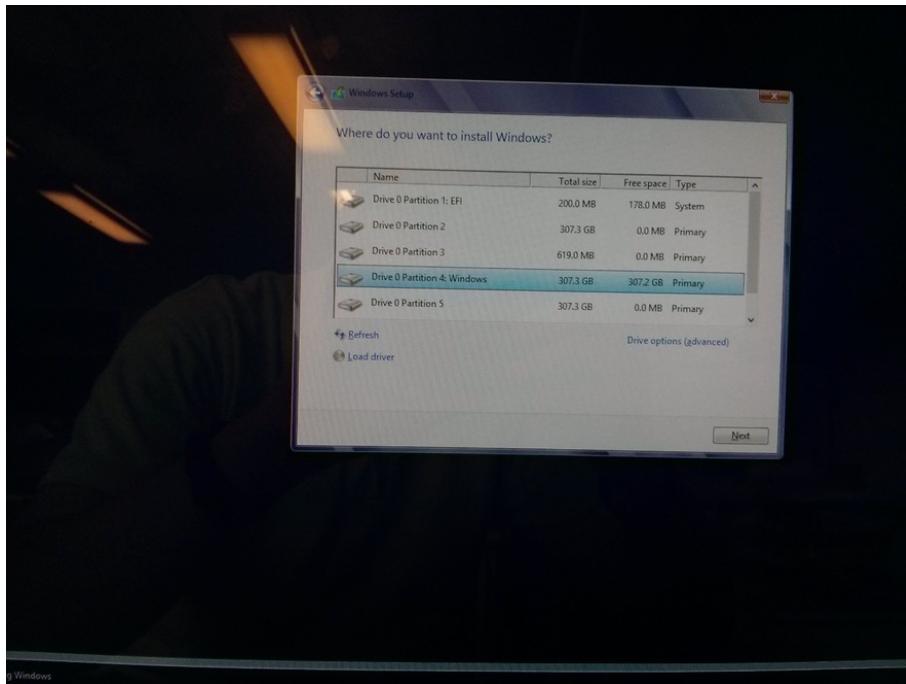
The next step is to select whether you are upgrading an existing installation of windows, or making a new install. We are again selecting a new, *Custom* installation.



Finally, Windows will ask you which partition you want to install on.

If you created a ntfs partition, and set a label, Windows will be able to read the label of this partition, unlike most of the other partitions. This will be a good indicator of which partition you want to install on.

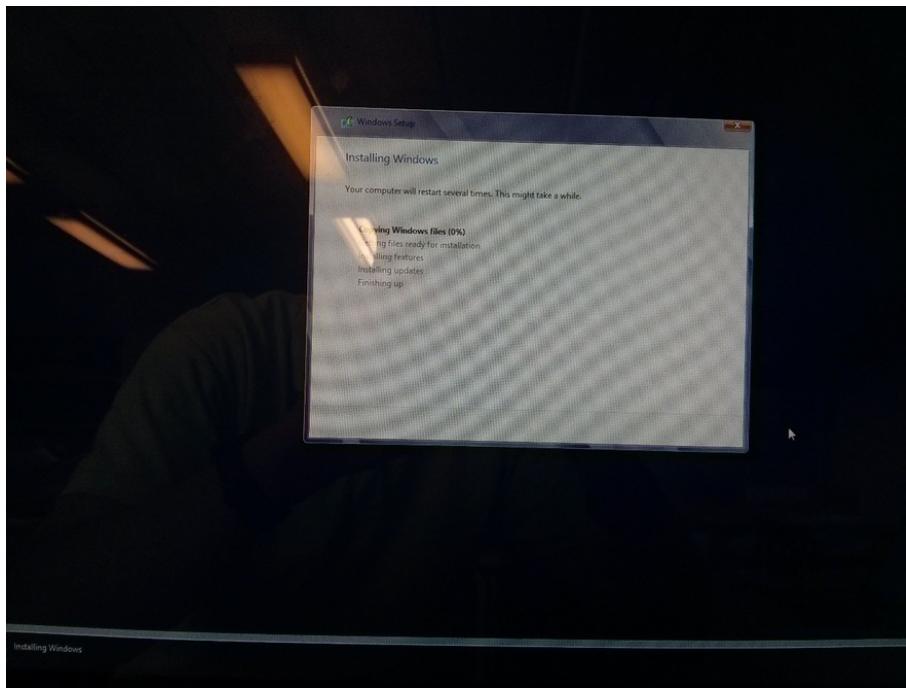
Otherwise you might need to recall which order you put the partitions in, to guess which one is the windows partition.



Select the correct partition here, and click *Next*.

That should be all the information Windows need from you to start the installation.

Once again you can have a little break, this will take a few minutes, probably around 5 - 15 in total.

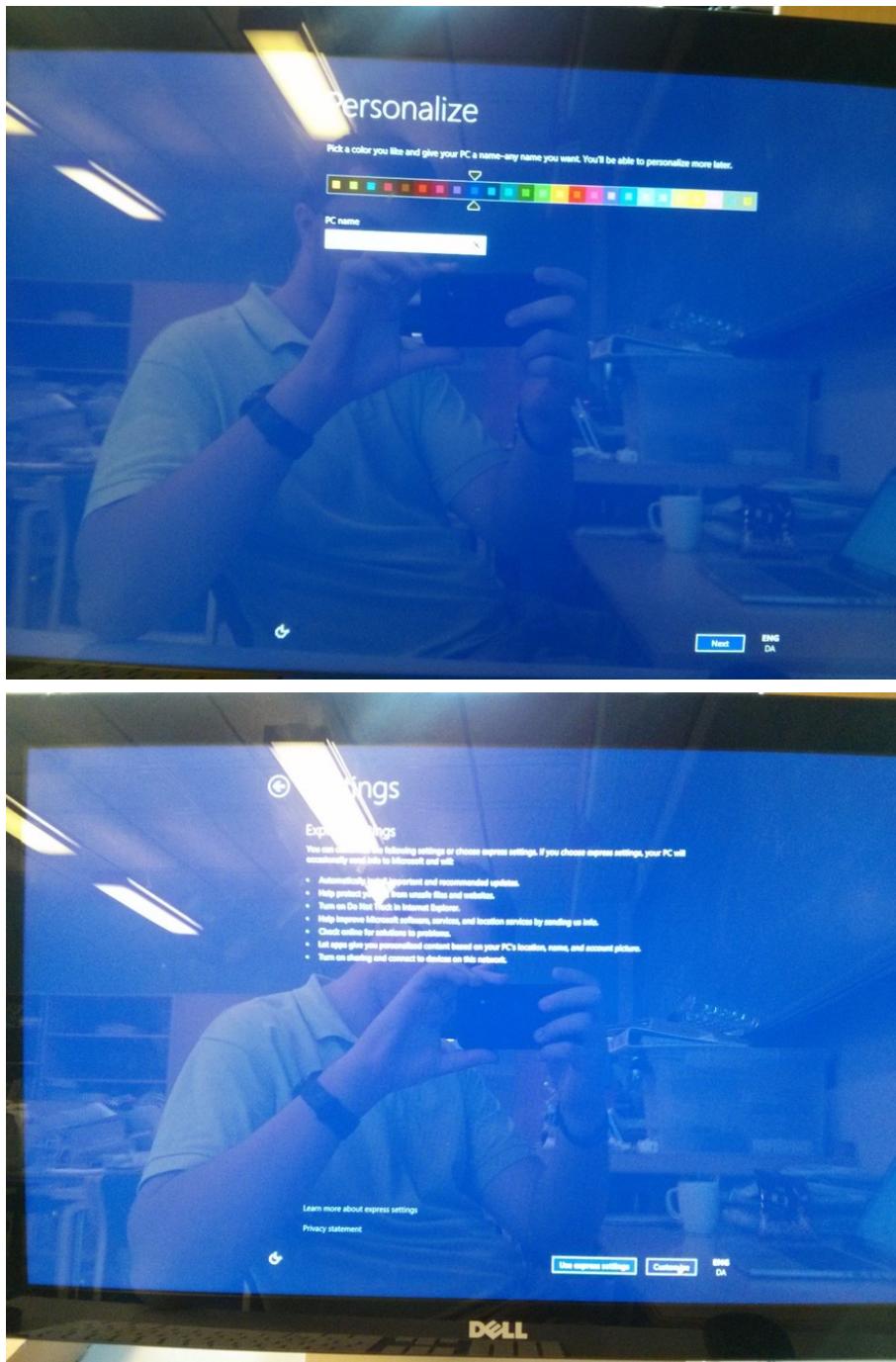


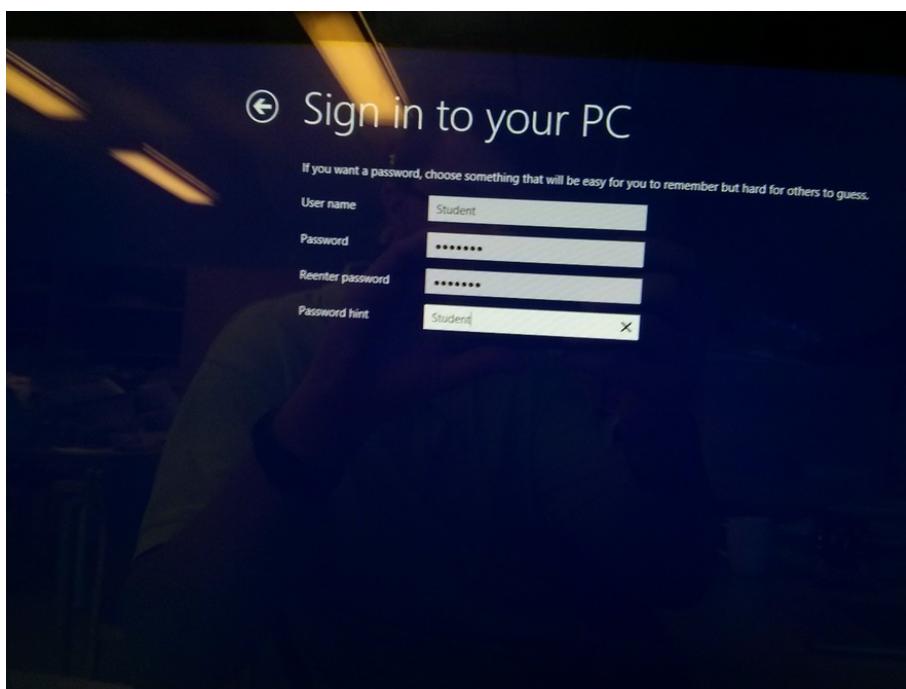
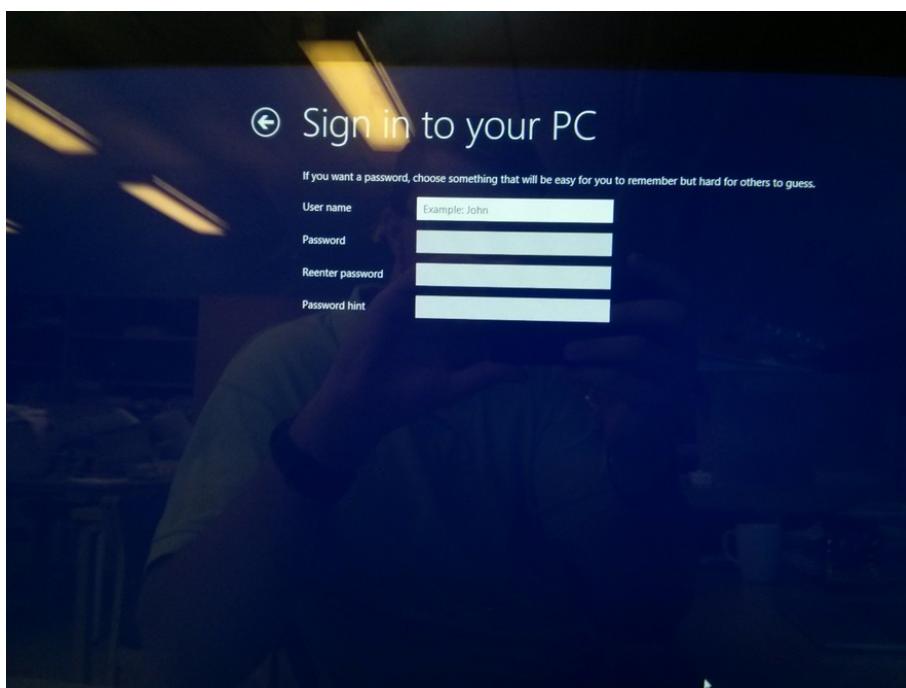
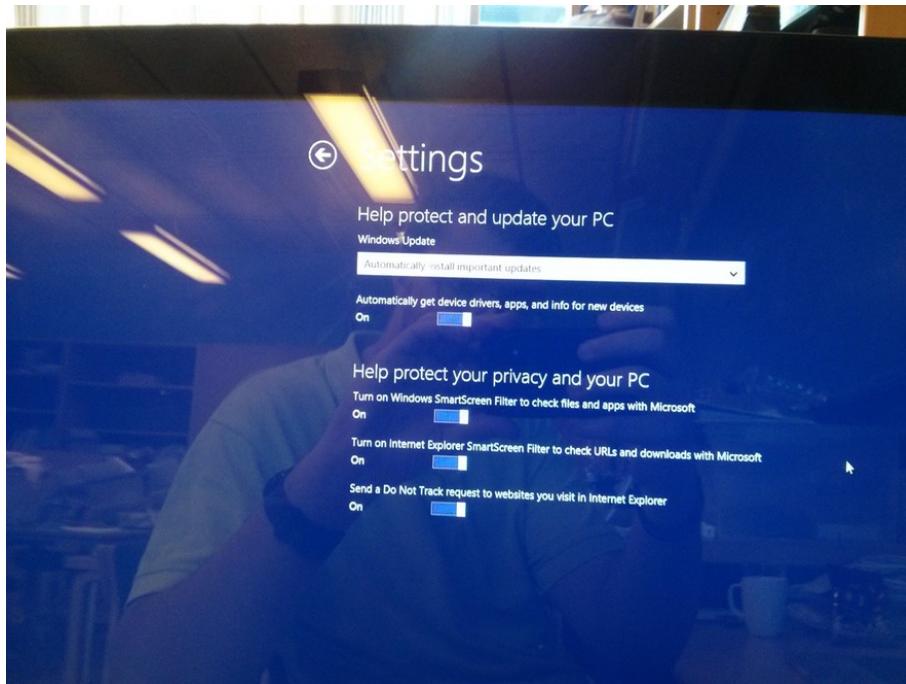
The computer may restart once or twice during the installation, don't worry, that's perfectly normal.

Once the installation is complete, you'll be asked to provide some personalisation settings.

Here you may select as you wish, but I will be doing the following.

- Color: Default
- PC Name: MediaLab-Macmini
- Settings: Customize
- Windows Update: Change to important only
- Username: Student







Again Windows will take a little while, this time to configure itself.

Once that's over, you should be booted into a working version of Windows.

However you may notice that the network will likely not work, and a number of other things may also be misbehaving.

This is because we need to install the drivers for the particular Mac, in Windows.

Technically you could go scavenging for these online, and find all the drivers.

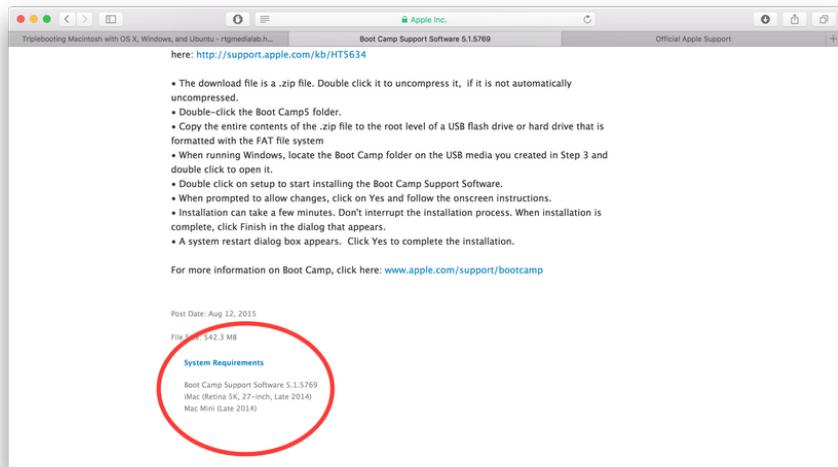
However Apple have made the job easy, and collected the drivers into packages they call *Boot Camp Support Software*. These packages can be downloaded off [Apple's support site](#).

Keep in mind that different versions of the software are required, depending on your specific version of Windows, and the particular hardware you are using.

For this part, go back to a computer with a working internet connection (You could even boot the Mac back into OS X, however I would recommend leaving it in Windows, and doing this on a different computer).

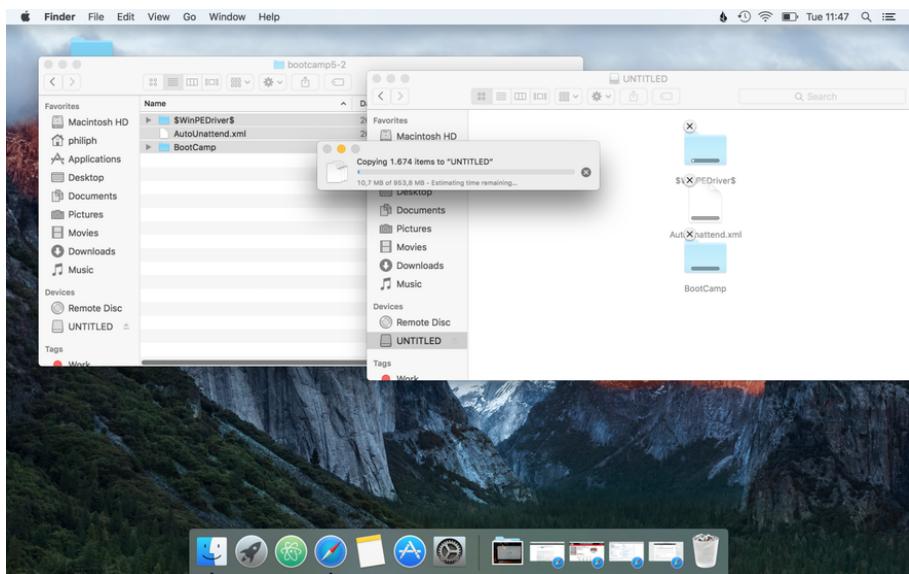
In my case I will need [Boot Camp Support Software version 5.1.5769](#), always check the *System Requirements* at the

bottom of the page, for supported hardware.



When you've found the right version of the software, go ahead and download then unzip it (If you are working on a Mac, this will be done automatically).

Grab a USB drive, or some other media, and copy the content of the downloaded folder onto it.



Once the drive is ready to go, remove it, and plug it into the Mac we're working with (Or if you opted to use the same computer, reboot, hold alt, and boot into Windows).

From Windows, open the drive.

You'll have a folder called *BootCamp*, within it will be a *Drivers* folder, and a *Setup* executable.

You could probably go ahead and try installing all the drivers yourself if you wanted, however the Setup program will handle

all that for us, so all we need to do is run it.

In my case I already have the program installed, so I'm offered to repair it, however from a fresh install you should be able to go through the installation program, simply by clicking *Next*, and all the drivers will be installed.

Once it's done however, we want to remove a program that Boot Camp also installs, this program allows you to reboot the computer into OS X from Windows. However with rEFInd we're about to install, we don't want that program present.

Go ahead and find the dialogue for uninstalling a program in your version of Windows.

In Windows 8 it's in *Control Panel > Programs > Uninstall a program*.

Here you'll find the program called *Boot Camp Services*. Go ahead and remove that.

Once it's removed, we're ready to handle the last step, rEFInd.

## Installing rEFInd

What is refind?

Refnd is the bootloader that will allow us to select which OS we want to boot into, when the computer starts. The default bootloader for mac does not support Ubuntu, and it requires us to hold down a special key to get the options, otherwise it just boots OS X.

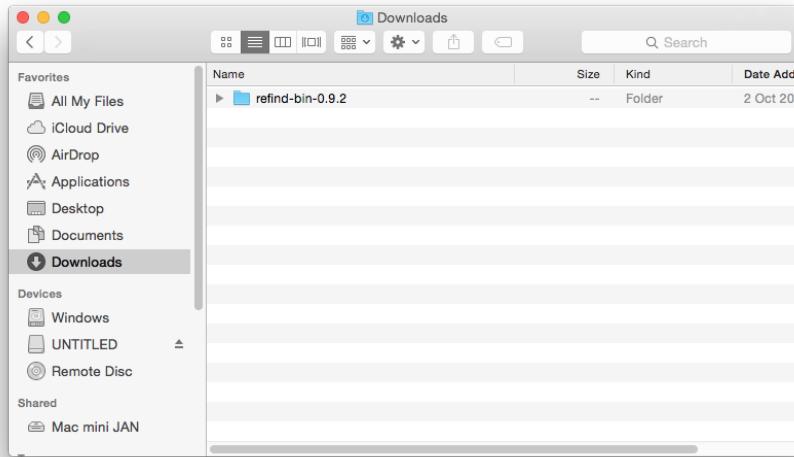
We want to be presented with a number of options, and then be able to select which one we want.

Although refind could be installed from Windows, the process is easier on OS X and Ubuntu, so to start off, we reboot the computer into OS X for this last step.

**Note** that installing windows also sets it as the default OS in the bootloader, so you will need to hold down the Option (alt) key when booting, to start OS X.

Once you're back in OS X, figure out where your refind folder is located (Downloading the zip version will automatically unpack it in OS X). In case you haven't gotten it yet, [go ahead and download it now!](#)

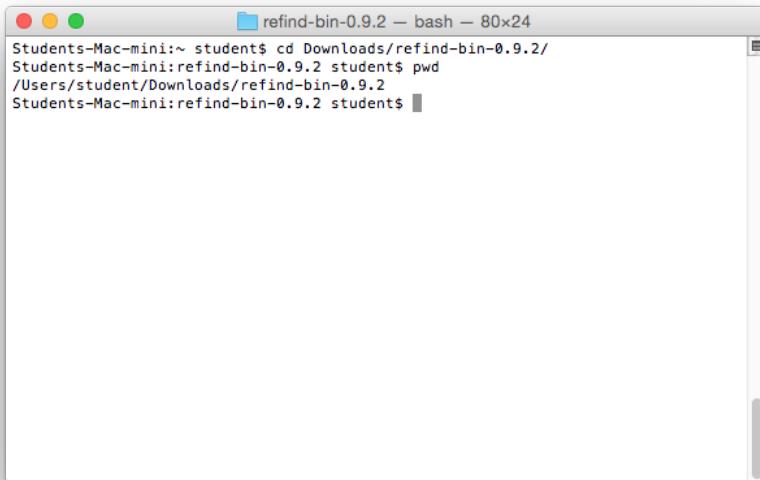
My refind folder is located inside *Downloads*.



Once you've figured out where it is, open a *Terminal* window, and navigate inside the folder.

As my refind is in *Downloads* and the version is 0.9.2 the path ends up being

`~/Downloads/refind-bin-0.9.2/`



Once there, execute the `install.sh` script, with the command  
`./install.sh`

**Note** the `./` in front of the name of the file, this is a syntax used in most UNIX based systems, to denote executing the file following.

You will be asked to provide the administrator password for the computer, do so.



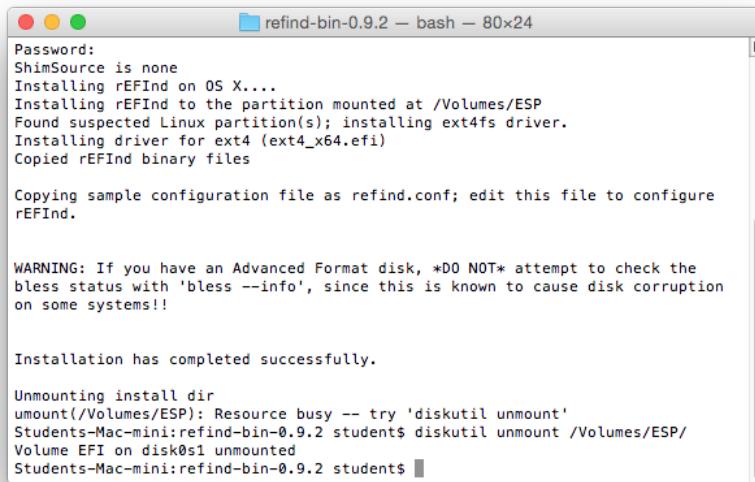
A screenshot of a Mac OS X terminal window titled "refind-bin-0.9.2 – sudo – 80x24". The window shows the command-line process of installing refind. The text in the terminal is as follows:

```
Students-Mac-mini:~ student$ cd Downloads/refind-bin-0.9.2/
Students-Mac-mini:refind-bin-0.9.2 student$ pwd
/Users/student/Downloads/refind-bin-0.9.2
Students-Mac-mini:refind-bin-0.9.2 student$ ./install.sh
Not running as root; attempting to elevate privileges via sudo...
Password: [REDACTED]
```

Once the program finishes, you may need to unmount the */Volumes/ESP* volume yourself, in that case, run the command the refind install script suggests

```
diskutil unmount /Volumes/ESP
```

Once it's done, refind should be installed.



A screenshot of a Mac OS X terminal window titled "refind-bin-0.9.2 – bash – 80x24". The window shows the completed refind installation process. The text in the terminal is as follows:

```
Password:
ShimSource is none
Installing rEFInd on OS X....
Installing rEFInd to the partition mounted at /Volumes/ESP
Found suspected Linux partition(s); installing ext4fs driver.
Installing driver for ext4 (ext4_x64.efi)
Copied rEFInd binary files

Copying sample configuration file as refind.conf; edit this file to configure
rEFInd.

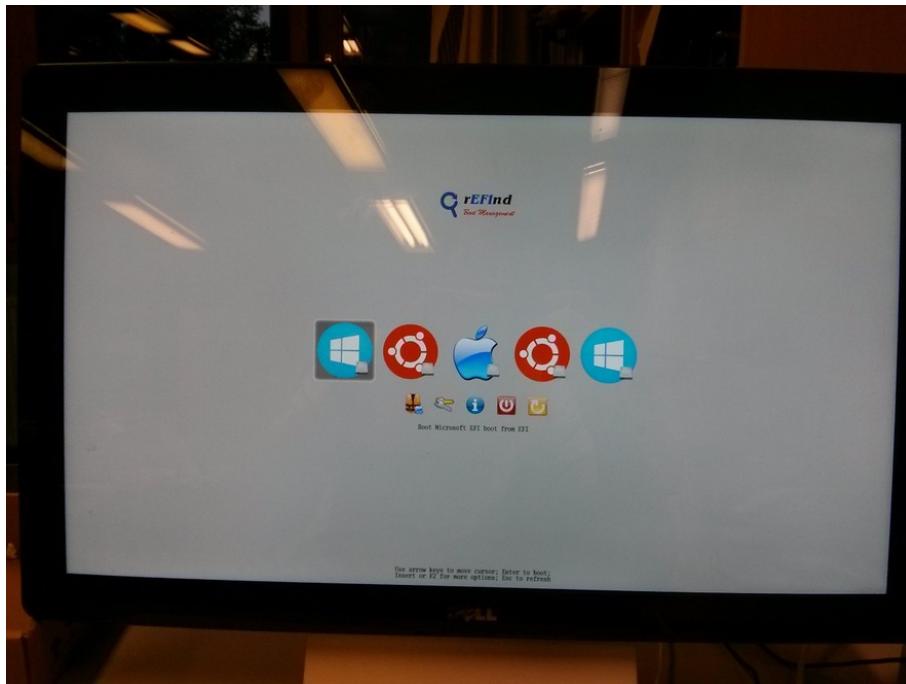
WARNING: If you have an Advanced Format disk, *DO NOT* attempt to check the
bless status with 'bless --info', since this is known to cause disk corruption
on some systems!!

Installation has completed successfully.

Unmounting install dir
umount(/Volumes/ESP): Resource busy -- try 'diskutil unmount'
Students-Mac-mini:refind-bin-0.9.2 student$ diskutil unmount /Volumes/ESP/
Volume EFI on disk0s1 unmounted
Students-Mac-mini:refind-bin-0.9.2 student$
```

To ensure it all works, try to reboot the computer.

If you are met with an image like this one, showing you Windows, OS X, and Ubuntu, you have succeeded!



There's a little bit of configuration work to be done with refind, but the basics are done.  
You've come very far!

## Configuring rEFInd (Optional)

You may stop at this point, and the computer would be perfectly useful.

However you may notice rEFInd gives us two more options than we have OSes installed. Ubuntu shows up both as the Ubuntu install itself, and the GRUB bootloader. That in itself may not be so bad, however the way it is configured, GRUB just boots us into Ubuntu anyway, we may want to be able to access GRUB in case of emergency, or disable it entirely.

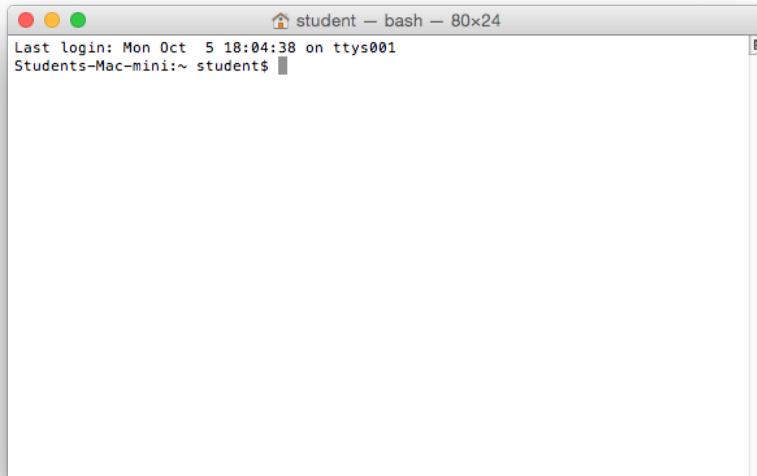
Windows appears to have two boot options, however one of them doesn't start anything, it simply says no startup device was found, we want to disable this option.

Finally, by default rEFInd will boot the previously booted OS 20 seconds after the computer starts, if nothing else is selected. We may want to change this to some other behavior, such as waiting for user input before booting anything.

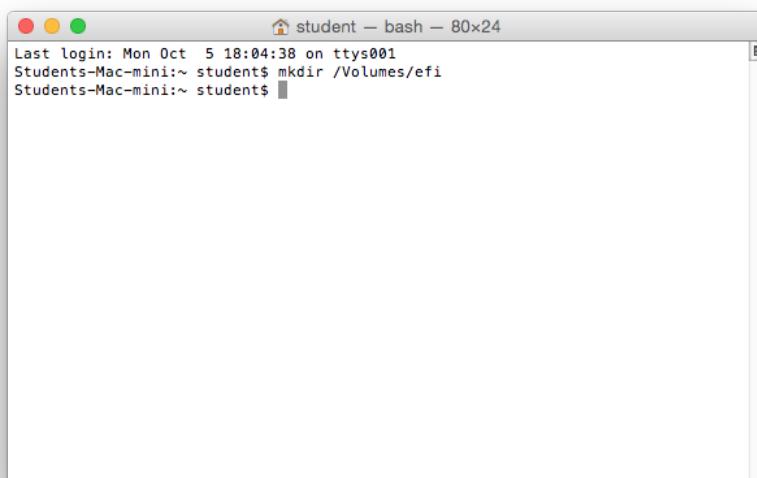
To do all of these things, we need to start by accessing the rEFInd configuration, for that, we need to mount the ESP (EFI System Partition), so that we can edit it.

I will be doing all of this in OS X, however you should in theory be able to access it from any system.

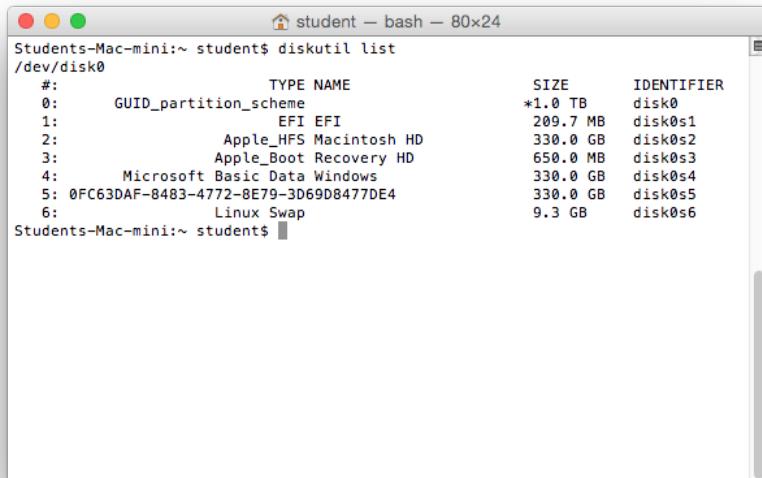
First off, open the terminal again.



Then create a temporary directory inside the *Volumes* directory, for our partition to be mounted, I will call mine *efi*  
mkdir /Volumes/efi



Then mount the partition, as you may remember from earlier it's the first partition on the disk. If you're unsure, you can double check with *diskutil list*

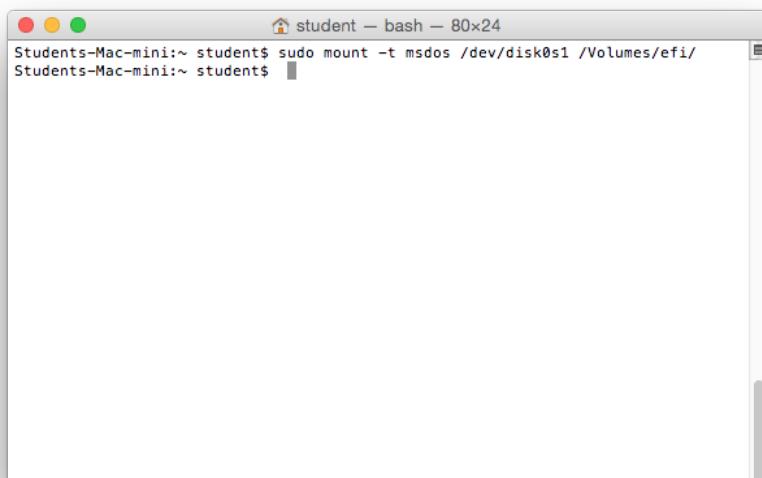


```
student - bash - 80x24
Students-Mac-mini:~ student$ diskutil list
/dev/disk0
      #:          TYPE NAME      SIZE    IDENTIFIER
      0: GUID_partition_scheme *1.0 TB  disk0
          #:          TYPE NAME      SIZE    IDENTIFIER
          1:   EFI   EFI           209.7 MB  disk0s1
          2:   Apple_HFS Macintosh HD   330.0 GB  disk0s2
          3:   Apple_Boot Recovery HD   650.0 MB  disk0s3
          4:   Microsoft Basic Data Windows   330.0 GB  disk0s4
          5: 0FC63DAF-8483-4772-8E79-3D69D8477DE4   330.0 GB  disk0s5
          6:   Linux Swap            9.3 GB   disk0s6
Students-Mac-mini:~ student$
```

As you can see, the identifier is *disk0s1*

We will mount that, as a fat partition (As that's the format it is), in OS X those are called msdos partitions.

```
sudo mount -t msdos /dev/disk0s1 /Volumes/efi
```



```
student - bash - 80x24
Students-Mac-mini:~ student$ sudo mount -t msdos /dev/disk0s1 /Volumes/efi/
Students-Mac-mini:~ student$
```

Now the content can be accessed at */Volumes/efi*. We navigate into */Volumes/efi/EFI/refind*, in here refind stores its configuration and other files.

Before we start editing, I will back up the configuration file for good order with the cp command.

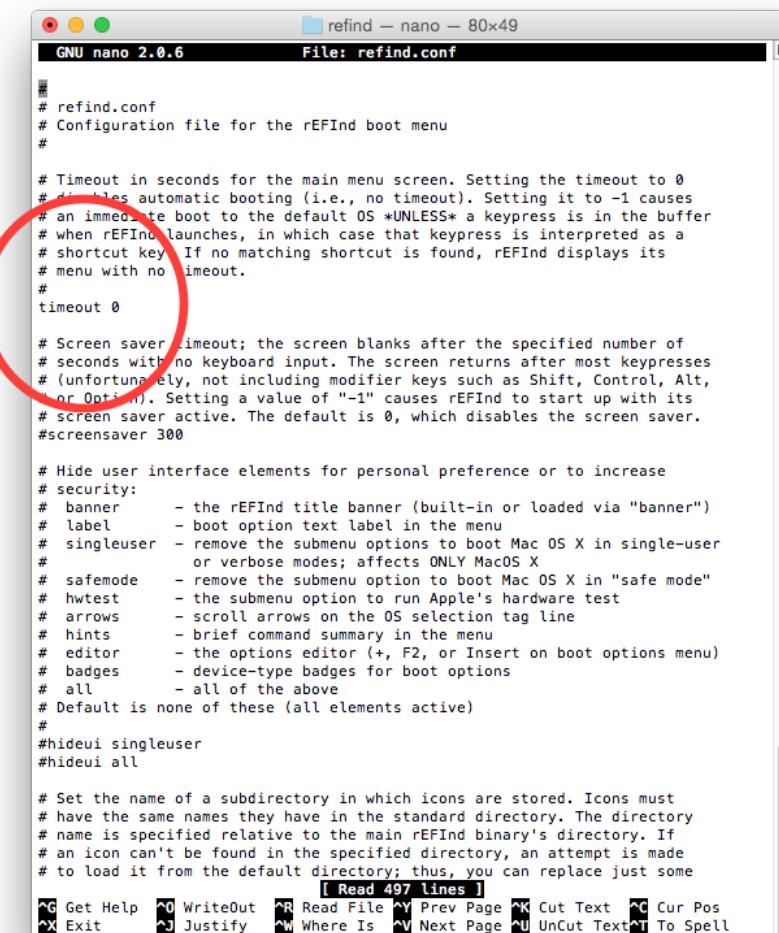
```
cp refind.conf refind.conf.default
```

Once that's done, we can edit the configuration file without fear of breaking everything.

**Note** however that if you make just the right screwup, you can still get the computer stuck in a position where rEFInd will give you no options to boot any OS, from any source, so do be careful.

Next up, removing the timeout.

This one is fairly easy, edit the refind.conf file with your favorite text editor (nano in my case), and replace the line "timeout 20" with "timeout 0", this will disable the timeout, meaning the computer will only ever boot an OS when the user selects one.



```
refind - nano - 80x49
GNU nano 2.0.6          File: refind.conf

#
# refind.conf
# Configuration file for the rEFInd boot menu
#

# Timeout in seconds for the main menu screen. Setting the timeout to 0
# disables automatic booting (i.e., no timeout). Setting it to -1 causes
# an immediate boot to the default OS *UNLESS* a keypress is in the buffer
# when rEFInd launches, in which case that keypress is interpreted as a
# shortcut key. If no matching shortcut is found, rEFInd displays its
# menu with no timeout.
#
timeout 0

# Screen saver timeout; the screen blanks after the specified number of
# seconds with no keyboard input. The screen returns after most keypresses
# (unfortunately, not including modifier keys such as Shift, Control, Alt,
# or Option). Setting a value of "-1" causes rEFInd to start up with its
# screen saver active. The default is 0, which disables the screen saver.
#screensaver 300

# Hide user interface elements for personal preference or to increase
# security:
# banner      - the rEFInd title banner (built-in or loaded via "banner")
# label        - boot option text label in the menu
# singleuser   - remove the submenu options to boot Mac OS X in single-user
#                 or verbose modes; affects ONLY Mac OS X
# safemode     - remove the submenu option to boot Mac OS X in "safe mode"
# hwtest       - the submenu option to run Apple's hardware test
# arrows       - scroll arrows on the OS selection tag line
# hints        - brief command summary in the menu
# editor       - the options editor (+, F2, or Insert on boot options menu)
# badges       - device-type badges for boot options
# all          - all of the above
# Default is none of these (all elements active)
#
#hideui singleuser
#hideui all

# Set the name of a subdirectory in which icons are stored. Icons must
# have the same names they have in the standard directory. The directory
# name is specified relative to the main rEFInd binary's directory. If
# an icon can't be found in the specified directory, an attempt is made
# to load it from the default directory; thus, you can replace just some
[ Read 497 lines ]
[G] Get Help [W] WriteOut [R] Read File [P] Prev Page [C] Cut Text [U] Cur Pos
[X] Exit [J] Justify [W] Where Is [N] Next Page [U] Uncut Text [S] To Spell
```

As for the boot options, I have opted for the option of disabling auto discovery of systems on the hard drive, and manually defining the boot options.

Navigate to the commented out line with the *scanfor* option, by default the computer will scan for internal systems among others. I want to keep all the other default options (Finding external USB drives, CDs etc.) but removing internal drives.

```

GNU nano 2.0.6          File: refind.conf
#
#scan_driver_dirs EFI/tools/drivers,drivers

# Which types of boot loaders to search, and in what order to display them:
# internal      - internal EFI disk-based boot loaders
# external       - external EFI disk-based boot loaders
# optical        - EFI optical discs (CD, DVD, etc.)
# netboot        - EFI network (PXE) boot options
# hdbios         - BIOS disk-based boot loaders
# biosexternal   - BIOS external boot loaders (USB, eSATA, etc.)
# cd             - BIOS optical-disc boot loaders
# manual         - use stanzas later in this configuration file
# Note that the last three require firmware support, which is
# not present on all computers.
# The netboot option is experimental and relies on the ipxe.efi and
# ipxe_discover.efi program files.
# On UEFI PCs, default is internal,external,optical,manual
# On Macs, default is internal,hdbios,external,biosexternal,optical,cd,manual
#
#scanfor external,biosexternal,optical,cd,manual

# By default, rEFInd relies on the UEFI firmware to detect BIOS-mode boot
# devices. This sometimes doesn't detect all the available devices, though.
# For these cases, uefi_deep_legacy_scan results in a forced scan and
# modification of NVRAM variables on each boot. Adding "0", "off", or
# "false" resets to the default value. This token has no effect on Macs or
# when no BIOS mode options are present via scanfor.
# Default is unset (or "uefi_deep_legacy_scan false")
#
#uefi_deep_legacy_scan

# Delay for the specified number of seconds before scanning disks.
# This can help some users who find that some of their disks
# (usually external or optical discs) aren't detected initially,
# but are detected after pressing Esc.
# The default is 0.
#
#scan_delay 5

# When scanning volumes for EFI boot loaders, rEFInd always looks for
# Mac OS X's and Microsoft Windows' boot loaders in their normal locations,
# and scans the root directory and every subdirectory of the /EFI directory
# for additional boot loaders, but it doesn't recurse into these directories.
# The also_scan_dirs token adds more directories to the scan list.

Get Help  WriteOut  Read File  Prev Page  Cut Text  Cur Pos
Exit      Justify    Where Is  Next Page  Uncut Text  To Spell

```

At this point, refind will find no systems on the computer for you automatically, now we need to add some options manually.

Navigate to the bottom of the file, here we will define the systems to boot, there's a few examples to help us find out how to do.

I have added the following options:

```

menuentry "OS X" {
    icon /EFI/refind/icons/os_mac.png
    volume "Macintosh HD"
    loader /System/Library/CoreServices/boot.efi
}

menuentry "Windows 8" {
    icon /EFI/refind/icons/os_win8.png
    loader /EFI/Microsoft/Boot/bootmgfw.efi
}

```

```

menuentry "Ubuntu" {
    icon /EFI/refind/icons/os_ubuntu.png
    loader /EFI/ubuntu/grubx64.efi
}

```



The screenshot shows a terminal window titled "refind - nano - 80x49" displaying the contents of the `refind.conf` file. The file contains several menuentry definitions for different operating systems:

```

menuentry "Windows via shell script" {
    icon \EFI\refind\icons\os_win.png
    loader \EFI\tools\shell.efi
    options "fs0:\EFI\tools\launch_windows.nsh"
    disabled
}

# Mac OS is normally detected and run automatically; however,
# if you want to do something unusual, a manual boot stanza may
# be the way to do it. This one does nothing very unusual, but
# it may serve as a starting point. Note that you'll almost
# certainly need to change the "volume" line for this example
# to work.
menuentry "My Mac OS X" {
    icon \EFI\refind\icons\os_mac.png
    volume "OS X boot"
    loader \System/Library/CoreServices\boot.efi
    disabled
}

menuentry "OS X" {
    icon /EFI/refind/icons/os_mac.png
    volume "Macintosh HD"
    loader /System/Library/CoreServices/boot.efi
}

menuentry "Windows 8" {
    icon /EFI/refind/icons/os_win8.png
    loader /EFI/Microsoft/Boot/bootmgfw.efi
}

menuentry "Ubuntu" {
    icon /EFI/refind/icons/os_ubuntu.png
    loader /EFI/ubuntu/grubx64.efi
}

```

The bottom of the terminal window shows a series of keyboard shortcuts:

- Get Help** (^G)
- WriteOut** (^O)
- Read File** (^R)
- Prev Page** (^Y)
- Cut Text** (^K)
- Cur Pos** (^C)
- Exit** (^X)
- Justify** (^J)
- Where Is** (^W)
- Next Page** (^V)
- Uncut Text** (^U)
- To Spell** (^T)

As a final change, I have disabled the Mok utility, as my users will not be needing it.

Navigating to the commented file with `showtools` I uncomment the line, remove the `mok_tool`, and enable the `apple_recovery` instead.

```
GNU nano 2.0.6          File: refind.conf

#
#use_graphics_for osx,linux

# Which non-bootloader tools to show on the tools line, and in what
# order to display them:
# shell           - the EFI shell (requires external program; see rEFInd
#                   documentation for details)
# memtest         - the memtest86 program, in EFI/tools, EFI/memtest86,
#                   EFI/memtest, EFI/tools/memtest86, or EFI/tools/memtest
# gptsync         - the (dangerous) gptsync.efi utility (requires external
#                   program; see rEFInd documentation for details)
# gdisk           - the gdisk partitioning program
# apple_recovery  - boots the Apple Recovery HD partition, if present
# windows_recovery - boots an OEM Windows recovery tool, if present
#                   (see also the windows_recovery_files option)
# mok_tool        - makes available the Machine Owner Key (MOK) maintenance
#                   tool, MokManager.efi, used on Secure Boot systems
# about           - an "about this program" option
# exit             - a tag to exit from rEFInd
# shutdown        - shuts down the computer (a bug causes this to reboot
#                   many users)
# reboot          - a tag to reboot the computer
# firmware        - a tag to reboot the computer into the firmware's
#                   user interface (ignored on older computers)
# netboot         - launch the ipxe.efi tool for network (PXE) booting
# Default is shell,memtest,gdisk,apple_recovery,windows_recovery,mok_tool,about$

# showtools shell, gdisk, memtest, apple_recovery, about, reboot, exit, firmware

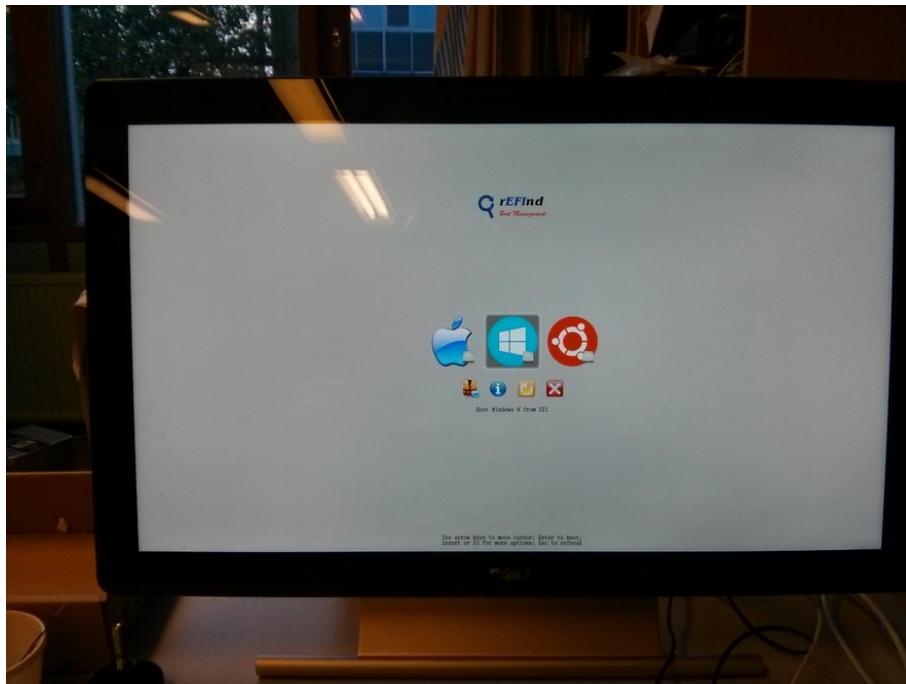
# Boot loaders that can launch a Windows restore or emergency system.
# These tend to be OEM-specific.
# Default is LRS_ESP:/EFI/Microsoft/Boot/LrsBootmgr.efi
#
#windows_recovery, files LRS_ESP:/EFI/Microsoft/Boot/LrsBootmgr.e

# Directories in which to search for EFI drivers. These drivers can
# provide filesystem support, give access to hard disks on plug-in
# controllers, etc. In most cases none are needed, but if you add
# EFI drivers and you want rEFInd to automatically load them, you
# should specify one or more paths here. rEFInd always scans the
# "drivers" and "drivers_{arch}" subdirectories of its own installation
# directory (where "{arch}" is your architecture code); this option
# specifies ADDITIONAL directories to scan.
# Default is to scan no additional directories for EFI drivers

[G] Get Help [W] WriteOut [R] Read File [N] Prev Page [K] Cut Text [C] Cur Pos
[X] Exit [J] Justify [W] Where Is [V] Next Page [U] Uncut Text [T] To Spell
```

Once all that is done, and you have ensured the boot loader settings in particular are set right, go ahead and save the file, then reboot the computer.

If you're lucky and everything worked out right, you should see a screen like this:



**Congratulations!** The job is done

## Configuration

---

### Windows

Set time and date settings to:

Start day - Monday

## Abbreviated process

---

1. Install OSX (If not already)
  - a. Country: Denmark
  - b. Keyboard: Danish
  - c. Network: MediaLab (log in)
  - d. Transfer: No
  - e. Apple ID: No
  - f. User:
    - Full name: Student
    - Account name: student (Auto generates)
    - Password: student
    - Hint: None
    - Picture: Math
    - Set time zone: Yes
  - g. Diagnostics: Allow
2. Resize the OS X partition inside Disk Utility
3. Use Ubuntu live to access GParted
4. Create partitions
5. Install Ubuntu
  - a. Language: English

- b. Download updates: Yes
  - c. Third party software: No
  - d. Install type: Something else
  - e. Install partition: /dev/sda5 as /
  - f. Location: Copenhagen
  - g. Keyboard: Danish - Danish
  - h. User:
    - Name: Student
    - Computer name: MediaLab-Mac
    - Username: student (Auto generates)
    - Password: student
    - Require password
6. Don't reboot
  7. Fix MBR
    - a. sudo gdisk /dev/sda
    - b. x
    - c. n
    - d. w
  8. Install Windows
    - a. Language: English
    - b. Time and currency format: English
    - c. Keyboard: Danish
    - d. Accept license
    - e. Install type: Custom
    - f. Partition: Windows
    - g. Settings: Custom
    - h. Disable automatic hotspots and contact shared networks
    - i. Ownership: I own it (Currently using local user accounts only)
    - j. Skip Microsoft account
    - k. User:
      - Username: Student
      - Password: student
      - Hint: Student
    - l. Discoverable: No
  - m. Open date and time settings:
    - Timezone: Copenhagen
    - First day: Monday
    - Short date: yyy-MM-dd
    - Long date: dddd, d MMMM, yyyy
    - Short time: HH:mm
    - Long time: HH:mm:ss

- Region: Denmark
  - Region settings - Aditional - Decimal "," Digit group ":"  
Measurement Metric
  - n. Language: Remove English keyboard option
9. Install BootCamp support software
  10. Log in to MediaLab WiFi
  11. Restart in OSX
  12. Install rEFInd
  13. Copy rEFInd config file
  14. Boot into Ubuntu
  15. Configure Ubuntu
    - a. Remove programs from menu bar
      - Amazon
      - Settings
      - Drives
      - Software center
    - b. Run apt-get update && apt-get upgrade
    - c. Set clock settings to 24H
    - d. Install WiFi Drivers - See [help.ubuntu](#)
    - e. Connect to WiFi
    - f. Disable guest user