

Ask Ubuntu works best with JavaScript enabled

Ubuntu Community Ask! Developer Design Discourse Hardware Insights Juju Shop More ~

sign up log in tour help

Ask Ubuntu is a question and answer site for Ubuntu users and developers. It's 100% free.

Here's how it works:

Sign up

Anybody can ask a question

Anybody can answer

The best answers are voted up and rise to the top

Creating a boot disc and Installing ubuntu ASUS X205T(A)

I am having problems with creating a boot disc and installing ubuntu on an ASUS X205T I just bought.

- I have successfully (I believe) created a few boot discs with 2 different programs and checked to make sure the burn process resulted in a boot disc with multiple files instead of one .iso.
- I have turned off the "secure boot control" and assigned my disc drive as the #1 "boot option priority."
- I have tried disabling the "windows boot manager" as the #2 priority and only have the external drive as a boot priority. It begins to read the disc and then brings me right back to the BIOS. If I enable the "windows boot manager" as the #2 priority it reads the disc and then carries on to windows OS.
- I have also tried booting from the external by using the windows GUI and selecting the specific drive as well as the other removable and DVD/CD options.

boot uefi asus

edited Feb 10 '15 at 17:01

 Antoine Orsoni  
170 7

asked Dec 12 '14 at 21:37

 Jason Childers  
6 1 1 2

Actually, I was able to get ubuntu to boot using a 32 bit grub with 64 bit operating system. The x205t isnt linux incompatible. Linux boots, but with a few bugs. I used a mix between 2 different tutorials. [jfwhome.com/2014/03/07/...](http://jfwhome.com/2014/03/07/...) [github.com/T100Ubuntu/T100Ubuntu/wiki/Install-to-internal-flash](https://github.com/T100Ubuntu/T100Ubuntu/wiki/Install-to-internal-flash) Wifi doesnt work still, dont try what is said in the tutorial, it puts the x205t in a boot loop and you wont be able to use ubuntu. – user362972 Dec 29 '14 at 22:24

3 Answers

I have been running Ubuntu 14.10 in a dual-boot configuration on my Asus EeeX205T for a couple of months now. Some things do not yet work – internal wifi/bluetooth, sound, power management – but with a usb wifi adapter, enough works, and works well enough, that I have used it as a travel machine -- a role in which it excels.

**UPDATE:** I am now running Ubuntu 15.04 on this machine, still dual-boot (though I haven't booted into Windows in months). **With 4.0 or later kernels, internal wifi now works and power management works in part (battery monitoring), but suspend, sound, and bluetooth still do not work.** With the 4.1 kernels, there are some occasional problems with freezing. Instructions for installing 15.04 should be essentially the same as below, but I have not tried installing it from scratch, and cannot verify that all of the steps are the same (or even necessary). Additional steps needed to enable internal wifi and other needed settings are included below.

Here is how I set it up to dual-boot Ubuntu 14.10. Note that much of the same process could be followed to set it up for an Ubuntu-only installation. Note also that you do have to boot into Windows a time or two:

PRE-INSTALLATION

1. In Windows, turn off BitLocker (this will allow Ubuntu to mount and/or resize the primary data drive):
  - In Windows, click on PC Settings/PC and devices/PC info
  - scroll to the bottom of the screen and click to turn off BitLocker/decrypt drive
2. Prepare Ubuntu Live bootable USB memory stick:
  - Download .iso file for 64-bit Ubuntu 14.10
  - Create a bootable memory stick:
    - In Windows, download and run Rufus

**Ask Ubuntu works best with JavaScript enabled**

window

- After the memory stick is set up, copy bootia32.efi to the /EFI/BOOT directory on the memory stick
    - Fetch bootia32.efi from github (?) or from one of the links listed at the end of this post.
    - Alternately, instructions for generating this file can be found here on github -- look for JF Well's site on the t100.
  - Boot Ubuntu Live:
    - Insert memory stick in Asus EEE USB port
    - In Windows, hold Shift key and click on Restart; choose Advanced Options/UEFI Firmware to enter BIOS
    - Turn off SecureBoot
    - Select USB stick as boot device
    - Save and exit
    - When grub menu loads, select Try Ubuntu
3. Optional but **highly recommended**: Make a backup of the complete SSHD image
- Plug in a USB external hard disk with at least 32GB of free space
  - Right-click on the OS and Recovery icons in the Unity launcher and choose Unmount for each
  - Open a terminal and run the following: `dd if=/dev/mmcblk0 of=/media/ubuntu /HD/FILENAME bs=1M`
    - (Replace HD with the name by which the external drive was mounted; replace FILENAME with the filename you wish to use for the resulting image (or set another path as desired).)
    - It will take about 20 minutes for the command to complete; do NOT interrupt it, or you will have to start over!
    - Note: to restore the X205T to its original state, you can run this command "in reverse": `dd if=/media/ubuntu/HD/FILENAME of=/dev/mmcblk0 bs=1M --` and yes, I have done this to verify that it works!
4. Optional: Resize Recovery partition (note that the Recovery partition is required for booting Windows on this machine, so it cannot be removed if you want dual-boot, but it can be reduced in size to allow some additional room):
- Right-click on the OS and Recovery icons in the Unity launcher and make sure each is unmounted (choose Unmount for each if needed)
  - Open a terminal and run `gparted`
    - Note: this runs very slowly at this point; be patient!
    - Use `gparted` to shrink the Recovery partition as small as it will go (a little less than 6 GB)
    - use `gparted` to expand the OS partition to take up the extra room
    - commit the changes (again, be patient ...)

**INSTALLATION (from Ubuntu Live)**

1. Set up a wireless connection (to download updates during installation):
  - Insert a USB based wifi adapter (Ubuntu does not yet recognize the internal wifi)
  - Choose a wireless connection and connect
2. Double-click on Install Ubuntu
  - Select language; continue
  - Click to install additional drivers and restricted software; continue
  - Choose Install Ubuntu alongside Windows; continue
    - Alternately, you can choose to replace Windows completely at this point
  - Set relative sizes of Ubuntu/Windows partitions as desired (I set mine up with 8 GB for Windows and 13 GB for Ubuntu); continue
  - Allow the installation to complete, but do not (yet) allow it to reboot. Before rebooting, open a terminal and enter the following commands:
    - `sudo -s //sets up a root shell`
    - `mount /dev/mmcblk0p5 /mnt // make sure that this is the partition to which Ubuntu was installed`
    - `mount /dev/mmcblk0p1 /mnt/boot/efi // make sure this is the EFI partition`
    - `for i in /dev /dev/pts /proc /sys /run; do sudo mount -B $i /mnt$i; done //transfer key elements of the current Ubuntu Live file system to the newly mounted Ubuntu partition on /mnt`
    - `mount -o bind /etc/resolv.conf /mnt/etc/resolv.conf // likewise make the current resolv.conf available via the mounted Ubuntu partition on /mnt`
    - `chroot /mnt /bin/bash //set up /mnt as the new root and run bash`

**Ask Ubuntu works best with JavaScript enabled**

- `apt-get install grub-efi-ia32 //... and install the 32-bit grub. (Note: Since 14.01 is discontinued, you may need to edit /etc/apt/sources.list and add http://old-releases.ubuntu.com/ubuntu/)`
- Reboot; the machine should boot to the grub menu, allowing a choice between Ubuntu and Windows. If so, congratulations – you have succeeded! If not, and if you made the recommended backup, see the note above for how to restore to the original state. If you did not make the backup ... you may have a brick on your hands.

**FINALIZE INSTALLATION**

## 1. Clean up Windows:

- Reboot to grub menu and choose Windows
- Allow Windows to check and “repair” drive C: (basically Windows is responding to the fact that the drive has been re-sized)
- When finished checking, it will automatically reboot; choose Windows again to verify that the Windows installation is still functioning properly.

## 2. Set up SDHC card reader:

- Reboot to grub menu and choose Ubuntu
- Open a terminal; run `sudo gedit /etc/modprobe.d/sdhci.conf`
  - note: this file may not yet exist; if not, create it
  - Add the following line: `options sdhci debug_quirks=0x8000`
  - Save and exit
  - In terminal, run `sudo update-initramfs -u -k all`
- Reboot; SDHC card reader should now work

**NEW: INSTALL LATEST KERNEL AND ENABLE WIFI/POWER MANAGEMENT/TWEAKS**

## 1. Download and install the latest kernel (4.0 stable, or 4.1RC6 unstable as of this writing):

- In Ubuntu, use browser to navigate to <http://kernel.ubuntu.com/~kernel-ppa/mainline/>
- Scroll to the bottom of the list; choose the latest kernel (or an earlier one if desired, especially if there are any bugs in the latest RC version)
- Download the following files (##### will be replaced with numbers indicating the kernel version):
  - `linux-headers-#####-generic-#####_amd64.deb`
  - `linux-headers-#####_all.deb`
  - `linux-image-#####-generic-#####_amd64.deb`
- Install these files by opening a terminal, changing to the download directory, and running `sudo dpkg -i linux*.deb` (doing it this way assumes no other linux\*.deb files are in the download directory)
- Note that if a kernel is not working properly or is no longer needed, it can be removed (each one takes up 200+ MB, so worth removing those not needed!):
  - get a list of the kernels installed: `sudo dpkg -l | grep linux-image-`
  - remove an image by running `sudo apt-get autoremove linux-image-#####-generic` (substitute the appropriate version numbers for #####)
  - obsolete packages can also be listed (`sudo dpkg -l | grep linux-` ) and removed (`sudo dpkg -P linux-#####` )
- Reboot; note that you can use the GRUB menu to select an earlier kernel to run if needed

## 2. Set up wifi (if latest kernel does not automatically do so; requires 4.0 kernel or later):

- Download `brcmfmac43340-sdio.bin` and `brcmfmac43340-sdio.txt` from internet (one source is on page 3 of third “helpful reference” below); copy these to `/lib/firmware/brcm` (requires `sudo`)
- `sudo gedit /etc/rc.local` and add the following lines:
  - `#rmmod brcmfmac`
  - `#rmmod brmutil`
  - `echo on > /sys/bus/platform/drivers/sdhci-acpi/INT33BB\:\00/power/control`
  - `modprobe brcmfmac`
- `sudo gedit /etc/modprobe.d/blacklist-custom.conf` (possibly creating the file) and add the following lines:
  - `blacklist brcmfmac`
  - `blacklist brmutil`

**Ask Ubuntu works best with JavaScript enabled**

lines added to rc.local above

- Reboot; internal wifi should now be working
3. Fix mmcblk0rmpb boot errors (if latest kernel does not automatically do so):
- See the fourth "helpful reference" above for changes needed to the `/lib/udev/60-persistent-storage.rules` file.
    - Alternately, scroll down the thread to find a post with the updated file; download it and save it in place of the original `/lib/udev/60-persistent-storage.rules` (backing the original up first is highly recommended!)
    - can also just save the updated file to `/etc/udev/60-persistent-storage.rules` (this should override the `/lib/udev` file)
  - Rebuild the initramfs: `sudo update-initramfs -u`
  - Optional: Add `rm -f /dev/mmcblk0rmpb` to `/etc/rc.local` (may help prevent system freezes)
  - Reboot; you should no longer see the `mmcblk0rmpb` boot errors. (This may also prevent other lags in the system)
4. Fix problems caused by attempts to suspend (if latest kernel does not fix the suspense problem):
- Goto System Settings/Power and choose "Do not suspend" for "Suspend when inactive for" (do this for both battery and AC)
  - `sudo gedit /etc/systemd/logind.conf` and add `HandleLidSwitch=ignore`
  - Reboot

At this point, you should have a mostly working dual-boot machine. What doesn't work yet (hopefully a new kernel down the road will fix these issues): \* Bluetooth (it sort-of sees the bluetooth, but not completely) \* Sound (volume control appears to work, but it is a "dummy" sound system) \* ACPI (power-saving, suspend)

What does work at this point (mostly): \* Internal wifi (if kernel > 4.0 and steps above are followed) \* USB ports \* SDHC card reader (if you do the tweak above) \* Screen brightness (from System settings, but not from the keyboard) \* Screen turn-off after specified time; lock when screen turns off - but note that sometimes if the system has locked after screen turn off, it will not return to the log-in screen when a key is pressed. In this case, it is possible to call up a terminal screen (`Ctrl` - `Alt` - `F1`), log in, and re-start the X server (`sudo restart lightdm`), but any open programs will be closed and any unsaved data may be lost. \* Built-in camera

**NEW: OTHER OPTIONAL SETUP**

1. Set up micro-SD card as `/home`:
- Insert a micro-SD card and format it to `ext4`
    - One way to do this is to install `gparted` (`sudo apt-get install gparted`); note that you will need a working wifi (see above!)
    - Note that you can also partition the micro-SD card and install one partition as `/home`, and another partition as something else
  - Copy all of the contents of `/home` to the SD card (or desired partition)
  - Determine the UUID of the SD card or the desired partition (`sudo blkid`) \* `sudo gedit /etc/fstab` and add the following line: `UUID=XXXX /home ext4 defaults 0 2` (replace XXXX with the UUID noted above)
  - Reboot
2. Make the Recovery partition non-mountable:
- `sudo blkid` to determine the UUID of the Recovery partition
  - `sudo mkdir /mnt/Recovery`
  - `sudo gedit /etc/fstab` and add the following line: `UUID=XXXX /mnt/Recovery ntfs noauto,umask=222 0 2` (replace XXXX with the UUID determined above)
  - Reboot

Continue setup of Ubuntu as desired!

**HELPFUL REFERENCES**

- <http://www.jfwhome.com/2014/03/07/perfect-ubuntu-or-other-linux-on-the-asus-transformer-book-t100/>
- <http://www.linuxforen.de/forums/showthread.php?276316-bootbaren-USB-Stick-F%FCr-32bit-UEFI-erstellen&p=1820224&viewfull=1#post1820224>
- <http://ubuntuforums.org/showthread.php?t=2254322&page=4>
- <https://bugs.launchpad.net/ubuntu/+source/systemd/+bug/1333140>

Ask Ubuntu works best with JavaScript enabled



Community

1



AHW

51

1

2

Could you check through your commands (I reformatted this to be readable, but didn't check all commands) - No Time Feb 11 '15 at 1:09

@ahw Awesome!.....I have few questions.....I want to install Ubuntu on my internal sd card. Possible? Internal sd card is not detected from Live USB? How do I partition manually instead of choosing "Install along side windows" ? Where do I install bootloader? And How do I know which one is efi?

- Khurshid Alam Sep 1 '15 at 5:42

@ahw If I go for manual partitioning what should I do with existing efi partition? Leave it as it is?

- Khurshid Alam Oct 3 '15 at 6:05

The short answer is, you can't do it. Your laptop isn't the only one. I recall reading the reason is an incompatibility with the linux kernel and the combination of a 64bit processor but a 32bit uefi like these laptops have. it's the baytrail processor and its stupid frankenstein of a uefi that's the problem. from memory the only way of getting linux to boot on one of these is to use a 64bit distro with a 32bit boot loader. so, long story short, you're stuck with windoze, which is about the only os that will boot.

the asus x205t isn't the only linux-incompatible computer out there. I had exactly the same incompatibly problems with a dell latitude 10 essentials and an asus transformer book t100, both of which also use baytrail cpus. it's unlikely ubuntu will ever install on these computers, at least not easily.

answered Dec 15 '14 at 1:24



senya

1

You may find your answer here:

<https://github.com/lopaka/instructions/blob/master/ubuntu-14.10-install-asus-x205ta.md>

I am not sure how great of a Stack-appropriate "answer" this is (on my part), but I am wrestling with this exact problem with my x205t right now. And I think this might be our best bet.

You apparently need to rewrite the GRUB loader manually when you're building your USB stick. This gets into some pretty dense bash command-line, so 'good luck'(?)

The tut's author (lopaka on GitHub) supposedly was able to make it work. I am still trying.

answered May 25 '15 at 20:07



Evan Exempt

1