**Dual-booting Ubuntu 14.10 on the Surface Pro 3**

15 December 2014



**Surface Pro 3 Intel i7 + 512GB running Ubuntu 14.10 Utopic Unicorn (Linux 3.17.6)**

**Prologue**

"Tablets? Why would I want one of those?"

...or so I'd been telling myself for sometime. Although there was always an allure to having a device for streaming media while lounging or on-the-go, I could never convince myself I actually needed to own a tablet. As a programmer, my Ubuntu desktop (as principally antiquated as it might be) covered my productivity needs at home, and when I travel, I would only occasionally touch my laptop, sticking to my phone to read e-mail and watch some YouTube. Why would I ever need another device to carry that falls somewhere between?

Enter the *Surface Pro 3*.

I had been keeping my eye on the Surface Pro lineage ever since it began, intrigued by its promise of hybrid capabilities: the concept of the tablet now seemed in reach. When I learned how the latest *Surface Pro 3* had matured, and that it was capable of dual-booting Ubuntu, that sealed the deal for me. A tablet I can finally justify by its capacity for productivity.

Here's how I got Ubuntu installed on my *Surface Pro 3*, and how you can do the same.

**UPDATE:**

Why Ubuntu? Why on a Microsoft Surface?

Following this article, I was asked numerous questions as to why someone would be crazy enough (or stupid, depending on who you ask) to do such a thing. Here are my reasons why I did it:

1. **I'm an open-source programmer**: I spend a lot of time developing on frameworks like Ruby and Rails, for which I prefer Linux/Ubuntu. On occasion, I dive back into the .NET/Windows world, or play a game, for which I prefer Windows.
2. **I want optimal performance**: Virtualizing Ubuntu on Windows 8.1 (via VMWare or Virtual Box) can be resource intensive, drain the battery, and otherwise perform sub-optimally. Running Ubuntu natively has been performant, quiet and cool; and I couldn't be happier.
3. **I want a laptop that can behave as a tablet when I need it to**: After doing a lot of research, the Surface Pro 3 seemed to do fill this role the best (as a matter of personal opinion.) And so I decided I'd use Windows as "tablet-mode", and Ubuntu acts as "desktop-mode" on my Surface Pro 3.
4. **The quirks don't bother me**: Missing a few buttons on a stylus I don't really use, or not being able to gesture aren't features that concern me, especially because I use "laptop mode" with a mouse (I hate touch pads.)
5. **Support is only going to improve:** Some of those quirks will be solved with the release of [Ubuntu Touch](http://en.wikipedia.org/wiki/Ubuntu_Touch). Others will be solved in the future Linux 3.19 kernel, meaning you won't have to compile your own drivers into the Linux kernel.
6. **It's a learning experience**: There's no better way to become a master of your own domain like getting down & dirty with a bootloader. I learned a lot writing this guide, which I hope will help save lots of time for enthusiasts reading this article who'd like to dual-boot their devices.

**TL;DR**

For those who need the short story, the process takes about 1 hour, and involves:

1. **Partitioning the SSD**   
   *10 minutes*
2. **Installing Ubuntu**   
   *20 minutes*
3. **Installing drivers**   
   *10 minutes*
4. **Installing the latest Linux Kernel (>= 3.19)**   
   *10 minutes*

(Alternatively, compiling your own Linux kernel with custom patches [much easier than it sounds], which takes ~1-2 hours, which is mostly compile time)

1. **Installing rEFInd as a replacement EFI boot loader *(OPTIONAL)***   
   *20 minutes*

At the end of the process, your Surface Pro 3 will boot into rEFInd, with options to load Windows or Ubuntu. The WiFi, Type Cover (both keyboard and touch pad), Touch Screen, and Bluetooth should work fine.

However there are some quirks while using Ubuntu with kernel 3.17.6:

* Touch pad multi-touch gestures do not work (e.g. scroll with two fingers, pinch to zoom, etc.)
* The N-Trig stylus can point, 'click' and drag, but the buttons (e.g. right-click) do not work.
* Volume buttons do not work.
* Sleeping the system while running Ubuntu does not work. (Instantly wakes up.)

NOTE: These quirks exist for kernel 3.17.6, and may or may not exist based on your drivers/patches. It's entirely likely that the latest Linux kernel will eliminate most of these quirks as driver support is added.

Despite these missing minor features in 3.19, overall, it works fairly well, especially if you dock your surface with keyboard, mouse and monitors to use it as a desktop.

Before continuing you should be comfortable around a Linux command line, or be brave enough to try, as reconfiguring the Surface to play well with Ubuntu requires a deep-dive into the software internals of the tablet. That said, it's all much easier than it sounds, so let's get going!

**What you need**

To make this work I would recommend:

* [**Surface Pro 3**](http://www.microsoftstore.com/store/msusa/en_US/pdp/Surface-Pro-3-i3-64-gb-tablet/productID.300190600?vid=304047100&WT.mc_id=rtm-Surface-Pro-3-i3-64-gb-tablet): I bought the Intel i7/512GB model, which came pre-loaded with Windows 8.1.
* **USB Thumb Drive (4GB+)**: For the Ubuntu Installer.
* **USB Mouse** (Optional)
* **USB Keyboard**: For typing before you finish installing drivers.

Then to connect these USB devices to the single USB port on the Surface Pro 3 you will also need:

* **USB Hub**: To connect all your devices simultaneously.
* **Charger**: So the battery doesn't die.

**-OR-**

* [**Docking Station**](http://www.microsoftstore.com/store/msusa/en_US/pdp/Surface-Pro-3-Docking-Station/productID.300191900) (recommended): Provides 5 USB ports, an additional Mini DisplayPort, ethernet, and charging. Ideal for extended use of the Surface as a laptop or desktop.

It's also possible to complete setup without the USB hub or docking station, but extremely frustrating. Without a doubt, it's definitely worth buying either of these before hand.

I would also recommend having an extra device on-hand, like a laptop or desktop, in case you need extra firepower when downloading/compiling software, or for reference while a browser may not be accessible on the Surface. My desktop proved valuable in this role.

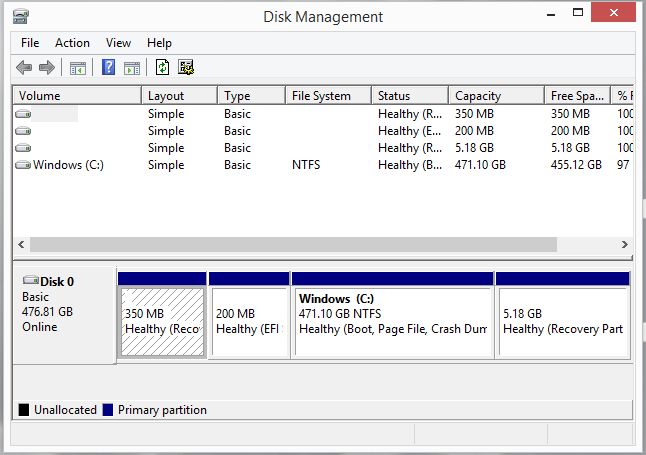
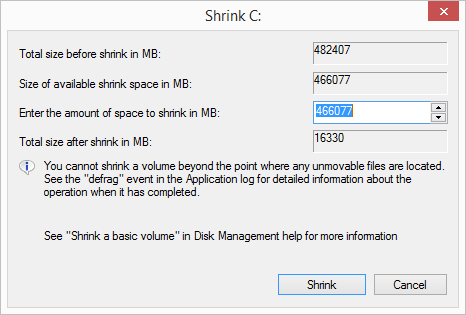
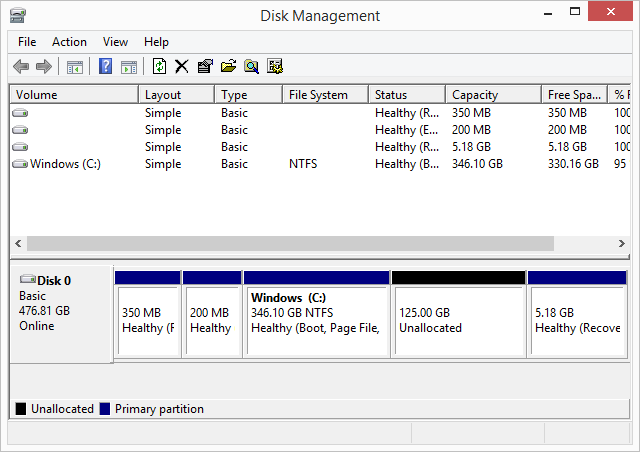
**Preparing for Dual Boot**

Boot your *Surface Pro* into Windows and follow your usual setup routine. Create your accounts, download your apps, etc... Follow-up this by running and completing all of your Windows updates to retrieve any firmware/other essential updates.

First order of business is to disable hibernation, as Linux cannot mount drives that are in hibernation. Open a *Command Prompt* as administrator and run:

powercfg -hibernate off

Then we need to create a partition for the new operating system, and a swap partition for paging.

1. Open the *Disk Management* utility via *Search*.   
   
2. Locate your main partition (usually *C:*). Right-click and select *Shrink Volume...*.   
   
3. In the dialog that opens, shrink the partition to free up enough space for Ubuntu and a swap partion. I would give it at least 24GB, but preferably 64GB+. (I gave it 128GB of my 512GB SSD.)   
   

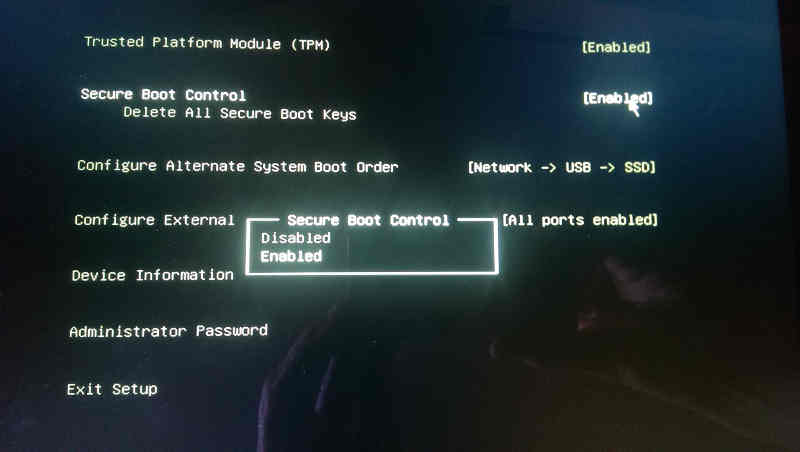
If you find that you cannot shrink sufficiently, the likely explanation is that your hard-drive is fragmented. It's a bit of a headache, but the following steps should help you defragment the SSD.

*Only follow these steps if you could not shrink the volume:*

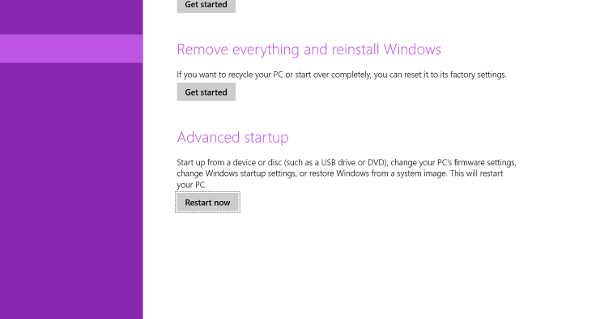
1. Disable *Windows Defender* and *BitLocker*. These complicate defrag operations.
2. Turn off the paging file (for now) via *Control Panel --> Computer --> Performance* menu.
3. Run the *Disk Cleanup* utility, including the option to cleanup system files.
4. Unfortunately the *Defragmenter* utility will not move system files around; we need to install a different utility to do this. Download and install *PerfectDisk*.
5. In the *PerfectDisk* defrag utility, for the *C:* drive, set the disk defrag options to "Prepare for Shrink" and "Move forward". Be sure it is also set to move system files (instead of ignore them.)
6. Run the defragmentation operation; you should see blocks move to the front of the disk, clearing space to shrink the *C:* drive.
7. Re-use the *Disk Management* utility to shrink the partition. If you still cannot sufficiently shrink the partition, you might need to play around with some of the *PerfectDisk* defrag settings; there are likely some blocks at the end of the partition that are preventing the volume from shrinking.

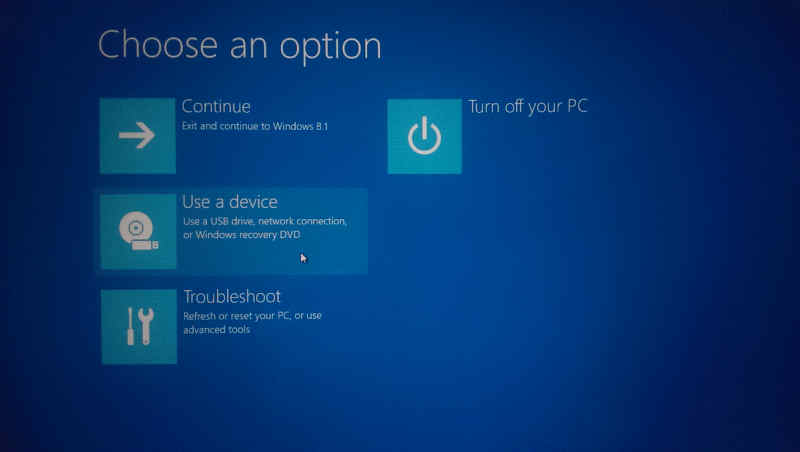
**Installing Ubuntu**

1. Power off your *Surface Pro*. Hold the "+" volume-key on the side of the device, then press the power button to load the EFI Boot menu.

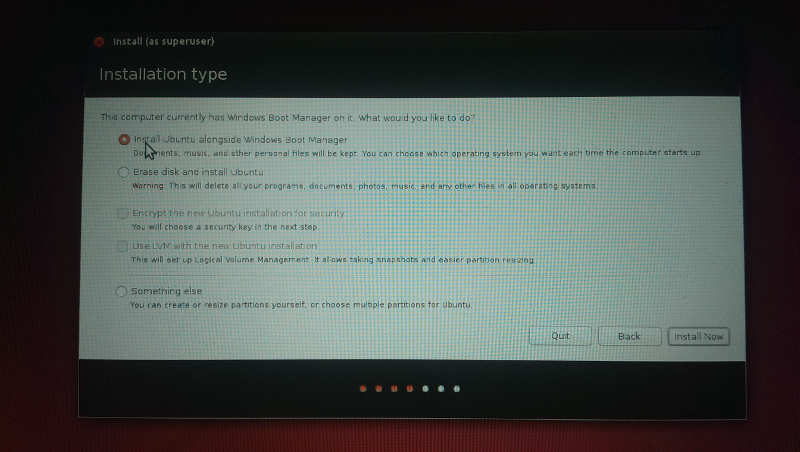


1. Select the *Secure Boot Control* option and set it to *Disabled*. This will permit Ubuntu to load (we'll cover this in more detail later, when I cover installing rEFInd.) Exit setup, and boot back into Windows.
2. Grab your USB thumb drive, wipe it clean, and download the latest Ubuntu Desktop ISO from the website. At the time of this writing, I used 14.10 Utopic Unicorn, but anything 14.04 Trusty Tahr and later should work fine.
3. Use the ISO to create a bootable USB following the instructions [here](http://www.ubuntu.com/download/desktop/create-a-usb-stick-on-windows).
4. Next, use the *Search* feature to open the *Recovery Options* menu. Select the *Restart Now* button at the bottom of the screen, and on the following menu choose to *Boot From Device* --> *USB Drive*, to reboot from the newly minted Ubuntu USB drive.



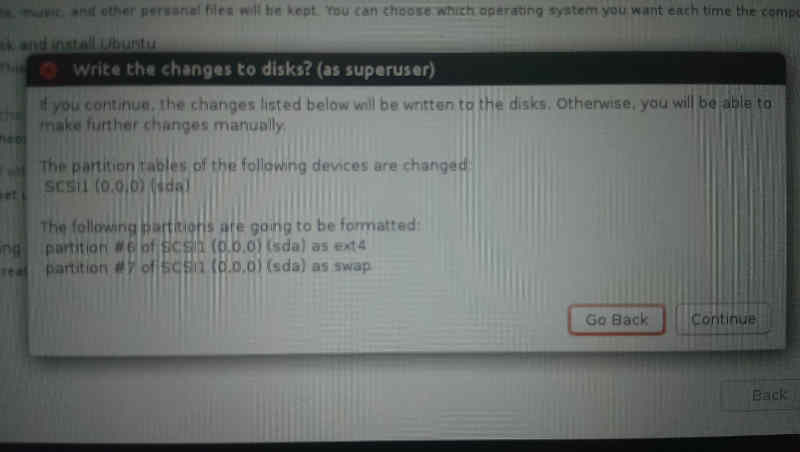


1. At this point, you should be presented with a boot screen that gives you the option to either "Try Ubuntu" or install it. I would recommend giving it a try first, to see how it feels on the Surface. (Note, you will discover that the Type Cover does not work; we'll fix that later.)
2. When you feel comfortable, kick-off the Ubuntu installer. When you reach the *Installation Type* prompt, you may or may not have a *"Install Ubuntu alongside Windows Boot Manager"*...



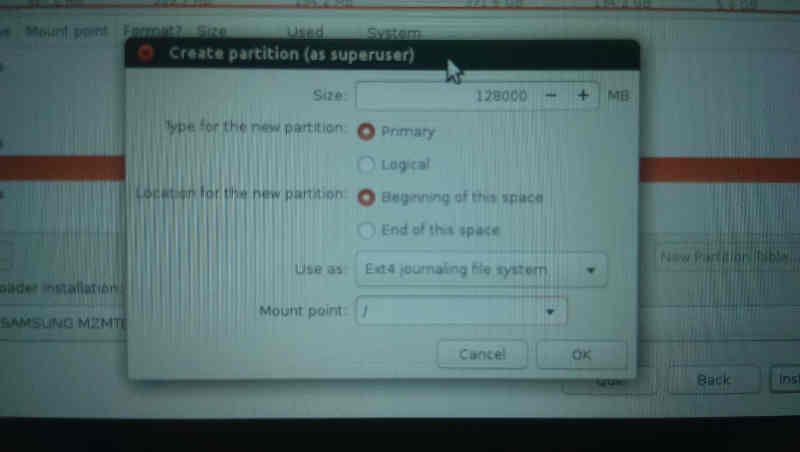
**If you do have the *"Install Ubuntu alongside Windows Boot Manager"* option:**

1. Then select the *"Install Ubuntu alongside Windows Boot Manager"* option, and confirm the settings in the pop-up that appears. It should automatically choose to partition the free space you created. Continue and complete setup.

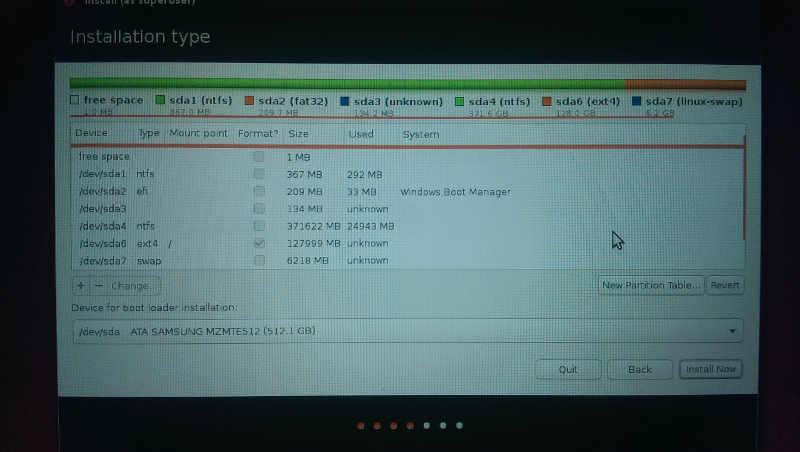


**If you do NOT have the *"Install Ubuntu alongside Windows Boot Manager"* option:**

1. Select *"Something else"*, in order to setup your partition.



1. From the un-allocated space you created by shrinking the Windows partition, create a partition for the Ubuntu OS (as ext4 mounted to '/'), and optionally (but recommended), a swap partition for paging. See [this post](http://askubuntu.com/questions/343268/how-to-use-manual-partitioning-during-installation) for more detail. Continue and complete the setup.



If you're unlucky like me, you might not have a USB Hub, where you can have both the USB Drive and keyboard plugged in. I used the *Character Map* keyboard to fill in the dialogs I couldn't type on, by tapping the [EN] symbol in the upper right corner, and copy-pasting the text into the forms. Super painful and slow... be smarter than me: get a USB hub.

**Installing Drivers**

At this point, Ubuntu should be installed and after rebooting, you should be presented with the GRUB bootloader: Load Ubuntu.

(If for some reason you're still getting forced into Windows, you can use your USB thumb drive as a bootloader to manually boot Ubuntu. See the *Manually Booting Ubuntu* section for help.)

First things you'll notice: your *Type Cover* keyboard doesn't work and the WiFi is unstable. Plug-in your USB keyboard, and let's get the WiFi stabilized:

1. Open a *Terminal* and install Git via sudo apt-get install git
2. Download the drivers from the Git repository:

git clone git://git.marvell.com/mwifiex-firmware.git

mkdir -p /lib/firmware/mrvl/

cp mwifiex-firmware/mrvl/\* /lib/firmware/mrvl/

Then let's add an entry to XOrg configuration to get the touch pad working on the Type Cover:

1. Open /usr/share/X11/xorg.conf.d/10-evdev.conf with *sudo* privileges using your favorite text-editor.
2. Add the following entry to the bottom of the file:

Section "InputClass"

Identifier "Surface Pro 3 cover"

MatchIsPointer "on"

MatchDevicePath "/dev/input/event\*"

Driver "evdev"

Option "vendor" "045e"

Option "product" "07dc"

Option "IgnoreAbsoluteAxes" "True"

EndSection

Once the Type Cover drivers are installed (in the next section), the touch pad should work after a system reboot.

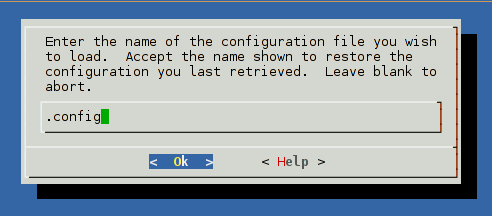
**Building the Linux Kernel**

***UPDATE: This section was necessary in the past, but no longer. You can skip this section entirely by*** [***downloading the latest Linux kernels***](https://www.kernel.org/) ***(3.19, 4.0+) instead.***

You'll notice the *Type Cover* isn't working; unfortunately the Linux kernel at this time does not natively include the necessary drivers. [The word on the street](http://www.reddit.com/r/SurfaceLinux/comments/2o6wta/will_linux_318_update_provide_typecover_support/) is that they'll be included in future kernel version 3.19. You can download a [binary kernel](http://surface3.rbel.co/kernel/) built with the drivers from [this repository](http://surface3.rbel.co/kernel/).

But if you don't trust these binaries, or want a fresher kernel version, we have to patch the Linux kernel with our own *Type Cover* drivers, until the official 3.19 Linux kernel is available.

1. Download the Source Code for the latest stable Linux kernel from [kernel.org](https://www.kernel.org/), and extract the contents into a directory.
2. Now we need to apply a patch for driver support. You can [download](http://cdn.delner.com/www/code/scripts/type-cover-3.patch) the file I created for 3.17.6, or you can create it in the root directory for the source code, as a new file called type-cover-3.patch
3. Add the following contents to the that file. NOTE: this patch was generated against kernel version 3.17.6 (as based on [this](http://www.spinics.net/lists/linux-usb/msg116763.html)), and isn't guaranteed to work against different versions.
4. diff -Naur linux-3.17.6/drivers/hid/hid-core.c linux-3.17.6-surface/drivers/hid/hid-core.c
5. --- linux-3.17.6/drivers/hid/hid-core.c 2014-12-07 14:48:01.000000000 -0500
6. +++ linux-3.17.6-surface/drivers/hid/hid-core.c 2014-12-13 18:37:00.902611695 -0500
7. @@ -702,6 +702,11 @@
8. if (((parser->global.usage\_page << 16) == HID\_UP\_SENSOR) &&
9. type == HID\_COLLECTION\_PHYSICAL)
10. hid->group = HID\_GROUP\_SENSOR\_HUB;
11. +
12. + if (hid->vendor == USB\_VENDOR\_ID\_MICROSOFT &&
13. + hid->product == USB\_DEVICE\_ID\_MS\_TYPE\_COVER\_3 &&
14. + hid->group == HID\_GROUP\_MULTITOUCH)
15. + hid->group = HID\_GROUP\_GENERIC;
16. }
17. static int hid\_scan\_main(struct hid\_parser \*parser, struct hid\_item \*item)
18. @@ -1857,6 +1862,7 @@
19. { HID\_USB\_DEVICE(USB\_VENDOR\_ID\_MICROSOFT, USB\_DEVICE\_ID\_MS\_DIGITAL\_MEDIA\_3K) },
20. { HID\_USB\_DEVICE(USB\_VENDOR\_ID\_MICROSOFT, USB\_DEVICE\_ID\_WIRELESS\_OPTICAL\_DESKTOP\_3\_0) },
21. { HID\_USB\_DEVICE(USB\_VENDOR\_ID\_MICROSOFT, USB\_DEVICE\_ID\_MS\_OFFICE\_KB) },
22. + { HID\_USB\_DEVICE(USB\_VENDOR\_ID\_MICROSOFT, USB\_DEVICE\_ID\_MS\_TYPE\_COVER\_3) },
23. { HID\_USB\_DEVICE(USB\_VENDOR\_ID\_MONTEREY, USB\_DEVICE\_ID\_GENIUS\_KB29E) },
24. { HID\_USB\_DEVICE(USB\_VENDOR\_ID\_MSI, USB\_DEVICE\_ID\_MSI\_GT683R\_LED\_PANEL) },
25. { HID\_USB\_DEVICE(USB\_VENDOR\_ID\_NTRIG, USB\_DEVICE\_ID\_NTRIG\_TOUCH\_SCREEN) },
26. diff -Naur linux-3.17.6/drivers/hid/hid-ids.h linux-3.17.6-surface/drivers/hid/hid-ids.h
27. --- linux-3.17.6/drivers/hid/hid-ids.h 2014-12-07 14:48:01.000000000 -0500
28. +++ linux-3.17.6-surface/drivers/hid/hid-ids.h 2014-12-13 18:35:17.262614087 -0500
29. @@ -647,6 +647,7 @@
30. #define USB\_DEVICE\_ID\_MS\_SURFACE\_PRO\_2 0x0799
31. #define USB\_DEVICE\_ID\_MS\_TOUCH\_COVER\_2 0x07a7
32. #define USB\_DEVICE\_ID\_MS\_TYPE\_COVER\_2 0x07a9
33. +#define USB\_DEVICE\_ID\_MS\_TYPE\_COVER\_3 0x07dc
34. #define USB\_VENDOR\_ID\_MOJO 0x8282
35. #define USB\_DEVICE\_ID\_RETRO\_ADAPTER 0x3201
36. diff -Naur linux-3.17.6/drivers/hid/hid-microsoft.c linux-3.17.6-surface/drivers/hid/hid-microsoft.c
37. --- linux-3.17.6/drivers/hid/hid-microsoft.c 2014-12-07 14:48:01.000000000 -0500
38. +++ linux-3.17.6-surface/drivers/hid/hid-microsoft.c 2014-12-13 18:39:28.386608291 -0500
39. @@ -274,6 +274,8 @@
40. .driver\_data = MS\_NOGET },
41. { HID\_USB\_DEVICE(USB\_VENDOR\_ID\_MICROSOFT, USB\_DEVICE\_ID\_MS\_COMFORT\_MOUSE\_4500),
42. .driver\_data = MS\_DUPLICATE\_USAGES },
43. + { HID\_USB\_DEVICE(USB\_VENDOR\_ID\_MICROSOFT, USB\_DEVICE\_ID\_MS\_TYPE\_COVER\_3),
44. + .driver\_data = MS\_HIDINPUT },
45. { HID\_BLUETOOTH\_DEVICE(USB\_VENDOR\_ID\_MICROSOFT, USB\_DEVICE\_ID\_MS\_PRESENTER\_8K\_BT),
46. .driver\_data = MS\_PRESENTER },
47. diff -Naur linux-3.17.6/drivers/hid/usbhid/hid-quirks.c linux-3.17.6-surface/drivers/hid/usbhid/hid-quirks.c
48. --- linux-3.17.6/drivers/hid/usbhid/hid-quirks.c 2014-12-07 14:48:01.000000000 -0500
49. +++ linux-3.17.6-surface/drivers/hid/usbhid/hid-quirks.c 2014-12-13 18:38:09.270610117 -0500
50. @@ -77,6 +77,7 @@
51. { USB\_VENDOR\_ID\_FORMOSA, USB\_DEVICE\_ID\_FORMOSA\_IR\_RECEIVER, HID\_QUIRK\_NO\_INIT\_REPORTS },
52. { USB\_VENDOR\_ID\_FREESCALE, USB\_DEVICE\_ID\_FREESCALE\_MX28, HID\_QUIRK\_NOGET },
53. { USB\_VENDOR\_ID\_MGE, USB\_DEVICE\_ID\_MGE\_UPS, HID\_QUIRK\_NOGET },
54. + { USB\_VENDOR\_ID\_MICROSOFT, USB\_DEVICE\_ID\_MS\_TYPE\_COVER\_3, HID\_QUIRK\_NO\_INIT\_REPORTS },
55. { USB\_VENDOR\_ID\_MSI, USB\_DEVICE\_ID\_MSI\_GT683R\_LED\_PANEL, HID\_QUIRK\_NO\_INIT\_REPORTS },
56. { USB\_VENDOR\_ID\_NEXIO, USB\_DEVICE\_ID\_NEXIO\_MULTITOUCH\_PTI0750, HID\_QUIRK\_NO\_INIT\_REPORTS },
57. { USB\_VENDOR\_ID\_NOVATEK, USB\_DEVICE\_ID\_NOVATEK\_MOUSE, HID\_QUIRK\_NO\_INIT\_REPORTS },
58. Apply the patch using:
59. patch -p1 --ignore-whitespace -i type-cover-3.patch
60. Install dependencies for the kernel compliation:
61. sudo apt-get install libncurses5-dev
62. sudo apt-get install kernel-package
63. Next setup the kernel configuration using your current config:
64. cp /boot/config-`uname -r` .config
65. make menuconfig



* Then at the makeconfig menu, press <ESC><ESC> to save and exit.
* And kick off the build via:
* make-kpkg clean
* fakeroot make-kpkg --initrd --append-to-version=-surface-pro-3 kernel\_image kernel\_headers
* Depending on your device, install time may take between 1-2 hours. Upon completion, two .deb files should appear outside the source code directory. Install them using:
* sudo dpkg -i linux-image\*.deb linux-headers\*.deb

**Installing rEFInd**

Now at this point things should be looking pretty good. The WiFi and Type Cover should be working with your new kernel, and the GRUB bootloader should be appearing, allowing you to choose between Ubuntu and Windows. Even the docking station ethernet port should be humming along. If you're happy with this, you can stop here.

However, there are two shortcomings that bothered me: 1) it auto-boots (after a delay) into Ubuntu when I don't have my keyboard active (a pain because I always want Windows in tablet mode, when I don't have a keyboard), 2) an ugly red boot-screen appears whenever you start the Surface Pro. Let's fix both of these.

The Surface Pro 3 is a UEFI-based system (as opposed to BIOS); one of the features enabled by default is called *Secured Boot*. This feature is designed to prevent untrusted bootloaders (e.g. malware) from hijacking the boot-sector of the hard-drive. In order to install our own bootloader (rEFInd), and have it be trusted, we must follow up our install by registering a *machine-owner key (MOK)* with UEFI on our system. We can then use this key to sign Linux kernels as trusted, so UEFI will permit them to boot.

For the purposes of this guide, we'll setup a trusted version of rEFInd based on [this guide](http://www.rodsbooks.com/refind/secureboot.html#shim):

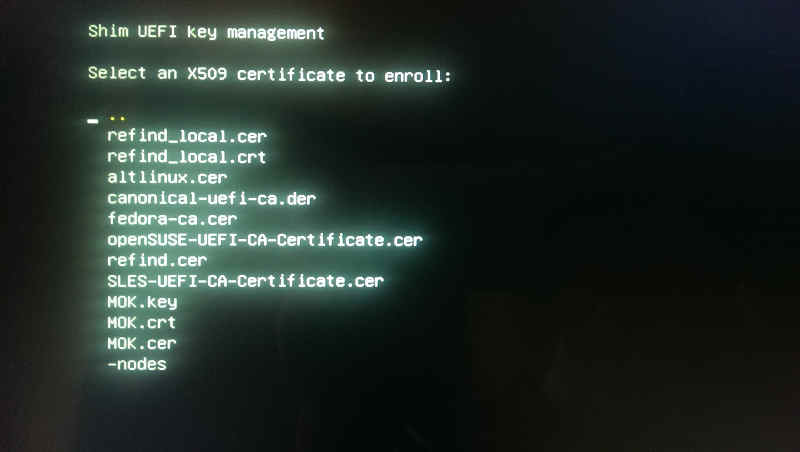
1. At this point, you should be able to reach Linux via GRUB. The latest GRUB bootloaders are trusted by Surface's UEFI, so we should be able to re-enable Secure Boot. Restart your Surface, reach the system settings screen, re-enable Secure Boot, and boot back into Linux via GRUB. (By enabling Secure Boot now, we will be avoiding some rEFInd install warnings.)
2. Download the [rEFInd binary ZIP package](http://www.rodsbooks.com/refind/getting.html) and extract the contents into a folder.
3. Download shim from [Matthew J. Garrett's download site](http://www.codon.org.uk/%7Emjg59/shim-signed/) and extract the contents (shim.efi and MokManager.efi) into the same directory as rEFInd.
4. Install rEFInd via sudo bash install.sh --localkeys --shim shim.efi
5. Verify that the bootloader has been registered using efibootmgr -v which should list an entry pointing at your EFI/refind/shim.efi
6. When you installed using the --local-keys option, Refind generated refind\_local.cer and refind\_local.crt files into the /boot/efi/EFI/refind/keys/ directory. It also copied these certificates to /etc/refind.d/keys/ along side a refind\_local.key file. This is your Machine Owner Key (MOK). We need to use this file to sign our Linux kernel.
7. Use the certificate and key to sign your Linux kernel, to make it Secure Boot safe:
8. sudo sbsign --key /etc/refind.d/keys/refind\_local.key --cert /etc/refind.d/keys/refind\_local.crt --output /boot/vmlinuz-3.17.6-surface-pro-3.signed /boot/vmlinuz-3.17.6-surface-pro-3
9. Double-check your /boot/efi/EFI folder. It should look something like:
10. /Boot
11. /bootx64.efi
12. /Microsoft
13. /Boot
14. /bootmgfw.efi (This is the default Windows bootloader)
15. /bootmgr.efi
16. ...
17. /refind
18. /grubx64.efi (This is actually rEFInd)
19. /shim.efi
20. /MokManager.efi
21. /refind.conf
22. /keys
23. /refind.cer
24. /refind\_local.cer
25. ...
26. /ubuntu
27. /grubx64.efi (This is the GRUB2 bootloader installed by Ubuntu)
28. /shimx64.efi
29. /MokManager.efi
30. /tools
31. **VERY IMPORTANT**: Make a backup of the Microsoft bootloader Microsoft/Boot/bootmgfw.efi! Copy it up a directory via mv bootmgfw.efi .. Sometimes GRUB might wipe the original out when it tries to hijack the bootloader for itself.

If a firmware update wipes out rEFInd (/Microsoft/Boot/bootmgfw.efi) or an enabled Secure Boot setting prevents rEFInd from loading, UEFI will find and load /Microsoft/bootmgfw.efi into Windows. Use *Recovery Options --> Advanced Startup --> USB Device* to jump into your backed up rEFInd or the Ubuntu installed GRUB. (If that fails, use a USB Boot Device and use the GRUB command line.)

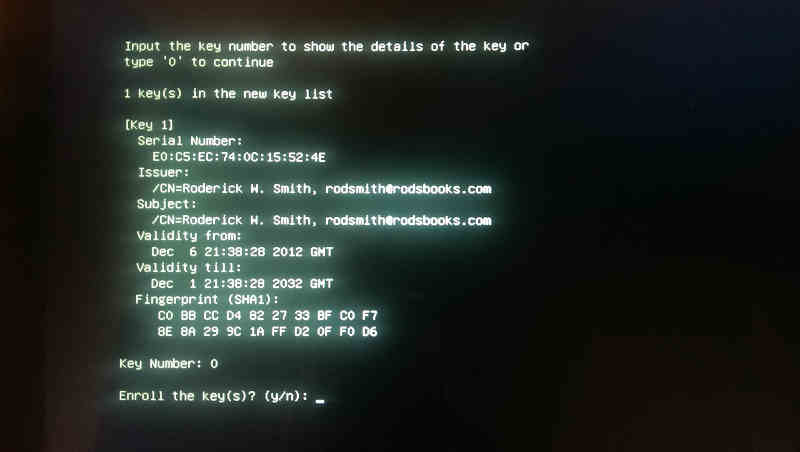
1. Reboot the system to load rEFInd. If rEFInd loads, you're done! Otherwise it might end up dumping you on the MokManager screen instead of into rEFInd or any other bootloader. This is the interface we'll use to register our MOK.

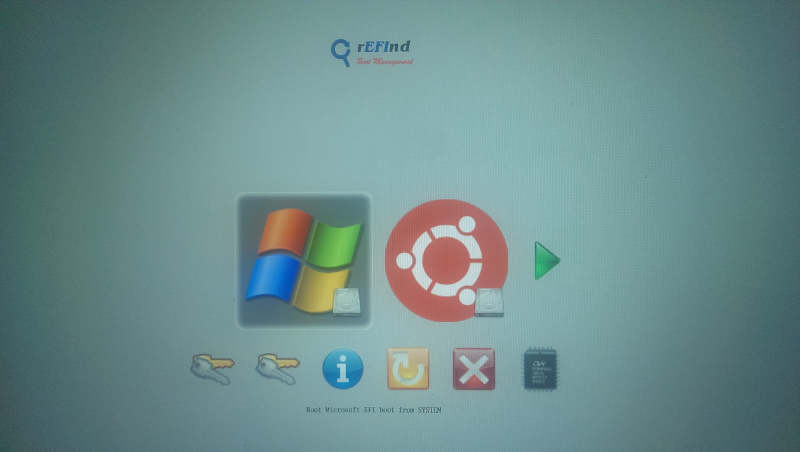


1. Tap down on the menu and select register key from disk.
2. Navigate down the ugly yellow directory structure to refind/keys, where you should find lots of \*.cer files.



1. Add the refind.cer, refind\_local.cer, and canonical-uefi-ca.der keys by selecting them then entering *"0"*. (Optionally, you can display more details about the key by entering *"1"*.)



1. Once you've added all of the keys, traverse back up the directory tree and select *"Continue boot"* at the main menu.
2. You'll now reach the rEFInd menu, where you should see an extensive list of boot managers and kernels to load. Select the one that has your surface-pro-3.signed kernel. 

**That's it!**

You can now play around with rEFInd settings to suit your tastes. I personally like setting Windows as the default with a short timeout, so I don't need my *Type Cover* when I want to use the Surface strictly as a tablet.

**Appendix: Manually Booting Ubuntu**

Use a bootable device (e.g. USB thumb drive) to load a GRUB command line. You can reach this command line by pressing 'c' when you're on the menu of a Ubuntu USB install (where you see "Try Ubuntu without installing.") In that command line prefixed grub>, use the following commands to boot Linux manually.

# To list partitions

ls

# Inspect the partitions to see which has the Linux installation

ls (hd0,2)/

# For example, if (hd0,2) has Linux installation...

# Set parameters and boot

set root=(hd0,2)

linux /boot/vmlinuz-3.17.6-surface-pro-3 root=/dev/sda2

initrd /boot/initrd-3.17.6-surface-pro-3

boot

**Appendix: If rEFInd gets overridden...**

If you install GRUB updates via Ubuntu's apt-get update or other package management system, you might find yourself being dumped back in GRUB after a reboot. To get things back to where they were, I suggest a rEFInd reinstall.

1. Make a backup copy of your refind.conf file, if you did any custom configuration.
2. Delete the /boot/efi/EFI/refind/ directory and all other rEFInd files.
3. Run efibootmgr -v to see the boot order, identify the rEFInd entry, then delete it via efibootmgr -b X -B where X is the boot number of the rEFInd entry.
4. Re-install rEFInd via the section above.

**Appendix: Upgrading to Windows 10**

If you have setup your Surface to dual boot previously using Ubuntu and Windows 8, you might be interested in taking the free upgrade to Windows 10. I personally performed the upgrade to Windows 10 using the tool Microsoft made available in Windows, and had no trouble. It did not modify my boot sector in any meaningful way, and I was able to continue dual-booting exactly as I had prior to the upgrade. To those considering the same, it should be a low-risk affair.

**Appendix: Resources around the Web**

* [rEFInd: EFI bootloader software](http://www.rodsbooks.com/refind/)
* [kernel.org: Linux kernels & source code](https://www.kernel.org/)
* [WinAero's "How to install Linux on Surface Pro 3"](http://blog.davidelner.com/dual-booting-ubuntu-14-10-on-the-surface-pro-3/)
* [/r/SurfaceLinux](http://www.reddit.com/r/SurfaceLinux/), [RubioJr's Surface Pro 3 thread](http://www.reddit.com/r/SurfaceLinux/comments/2bhfk5/ubuntu_1404_on_the_surface_pro_3_thread/), and [the thread for this post](http://www.reddit.com/r/SurfaceLinux/comments/2pbcwj/stepbystep_guide_for_install_ubuntu_on_the/).
* [Ubuntu Forum Surface Pro 3 thread](http://ubuntuforums.org/showthread.php?t=2231207)
* [Linux kernel patch for Surface Pro 3 drivers](http://www.spinics.net/lists/linux-usb/msg116763.html)