# <u>How can I mount my iPhone 6s on Ubuntu 16.04?</u>

https://askubuntu.com/questions/812006/how-can-i-mount-my-iphone-6s-on-ubuntu-16-04

## **Option 1: Using a script**

If you want to save yourself some time, you can download a script <u>here</u> to do most of the work of the process for you. Once downloaded, you will need to change the permisions so you can execute the script. Assuming you downloaded it with the default name, iphone\_setup.sh, cd to the directory in which you downloaded the file and do chmod u+x iphone\_setup.sh

Convert the Windows line endings by doing ex -bsc '%!awk "{sub(/\r/,\"\")}1"' -cx iphone setup.sh

Then run the script with root privileges using sudo ./iphone setup.sh

This will complete all of **Step 1** of the manual setup for you, as well as **Step 3** and **Step 4**. You will then need to do **Step 2** and **Step 5** of the manual setup after the script finishes running.

#### **Option 2: Doing it manually**

#### Step 1: Installing the tools

Before plugging in the iPhone, you will need to install the several programs to make it possible to mount the iPhone.

#### Step 1.1: Installing several important tools with apt-get

Do the following in the terminal to install a few packages that will be needed for any version of iOS. sudo apt-get install ideviceinstaller python-imobiledevice libimobiledevice-utils python-plist usbmuxd

If you are connecting an iPhone with an iOS version before iOS 9, you can skip the remaining substeps of step 1 and instead just do the following: sudo apt-get install libimobiledevice6 libplist3 ifuse

#### Step 1.2: Installing tools for building

Use apt-get to install a few programs needed to build the programs in the following steps sudo apt-get install libtool autoconf automake

#### Step 1.3: Installing libplist

First, install the required dependencies for building libplist. In order to do this, do the following: sudo apt-get install libxml2-dev python-dev

Then download the <u>latest version</u> of libplist from GitHub, and extract the contents of the zip file to some directory. For instance, if you are in the directory where you downloaded the libplist zip file, do unzip libplist-master.zip.

You should now have a directory called "libplist-master" in the directory to which you extracted the libplist zip file. cd into this directory from the terminal, and the run ./autogen.sh

When the ./autogen.sh script is done running, run make

And, finally, run sudo make install

## Step 1.4: Installing libusbmuxd

This step is similar to the previous step, except we are installing libusbmuxd instead of libplist. First, download the <u>latest version</u> of libusbmuxd from GitHub. Again, extract the contents to a directory, and cd to the directory libusbmuxd-master. Then run the following: ./autogen.sh

When this is finished, run make

followed by sudo make install

# **Step 1.5: Installing libimobiledevice** First, install the build dependencies by doing the following: sudo apt-get install libssl-dev

Then download the latest version of libimobiledevice from GitHub.

Extract as in the previous two steps; you should get a directory inside the directory to which you extracted called libimobiledevice-master. cd into this directory, and, again, run ./autogen.sh

When this is finished, run make

followed by sudo make install

#### Step 1.6: Installing a better version of usbmuxd

First, uninstall the old version of usbmuxd by doing sudo apt-get remove usbmuxd

Then, install the build dependencies by doing sudo apt-get install libimobiledevice-dev libplist-dev libusb-dev libusb-1.0.0-dev libtool-bin libtool

Then, download the <u>latest version</u> of usbmuxd from GitHub. Extract and cd to the usbmuxd-master directory. Again, run ./autogen.sh

When this is finished, run make

followed by sudo make install

# Step 1.7: Installing ifuse

This is the last thing you will need to install! First install, the build dependencies by doing sudo apt-get install libfuse-dev

Download the latest version of ifuse from GitHub. Extract it to some directory, and cd into the directory ifuse-master, and cd into that directory.

This time there is an extra step in building the program. Do ./autogen.sh

as usual, but then do ./configure

as well. Then, continue on to the normal make

and sudo make install

# Step 2: Running usbmuxd and attaching iPhone

This step is simple. Run usbmuxd in the terminal, and then plug in the iPhone. Now check to see if the device was recognized correctly by doing dmesg | grep ipheth

If nothing shows up, try disconnecting the iPhone, running usbmuxd again, and then plugging back in. Then check again.

#### Step 3: Creating a mount point for the iPhone

You can manually create a mount point for the iPhone by doing sudo mkdir /media/iPhone

You will then likely want to change the permissions for the mount point. Do sudo chmod 777 /media/iPhone

## Step 4: Editing the ifuse configuration file

The ifuse configuration file /etc/fuse.conf requires editing if you want to access the iPhone without being root. Edit the configuration file using your favorite editor, for example gedit sudo gedit /etc/fuse.conf

In the file ensure that the following two lines are under the line that says # Allow non-root users to specify the allow\_other or allow\_root mount options:

op\$ user\_allow\_other

Save the file and quit the editor.

#### Step 5: Pairing the iPhone

Run the following line in order to pair your iPhone using idevicepair: idevicepair pair

#### Step 6: Mounting with ifuse

Run the following line to mount the device at the mount point specified earlier: ifuse /media/iPhone

#### NOTE:

At this point you may mount the root filesystem if you have your phone jailbroken by doing the following line instead ifuse /media/iPhone/ --root

The iPhone should now be accessible at /media/iPhone through your file browser.

When you want to unmount, do the following two lines fusermount -u /media/iPhone/ idevicepair unpair

These steps were adapted for xenial from this tutorial at dedoimedo, then further modified to suit devices with iOS 9+.