Building customised Raspberry Pi
images for the NOOB system

Purpose:

The purpose of this guide is to help anyone wishing to build and deploy their own customised Pi, using the Noobs system. This might include teachers / network admins and anyone else who might wish to make lots of SD cards in a short space of time.

Noobs?

The Noobs (New Out Of Box System) makes deploying Raspberry Pi builds much faster and enables users to roll back their set up if mistakes are made. I'm a teacher and recently started using Noobs to provide students with Pi build that was preconfigured ready for use on our network. Rather than spend 10-15 mins writing a traditional .img file I can prep an SD card in about a minute.

Currently if you customised an image direct from the raspberry Pi downloads site, it will not be compatible with the way Noobs works with images. Therefore you will need to convert to a suitable format.

What will I need?

There may be other ways to achieve this but this list reflects how I converted my custom images.

- A customised Pi build (on sd card)
- SD Card reader
- Linux machine (might be able to do this from a Pi, with some extra storage)
- anywhere between 10 minutes and a hour depending on the size of the image.
- The latest noobs software (1.2.1) from Pi downloads section

You might be able to achieve this on a windows machine but I'm not sure what software you'd need.

STEP 1 – Download and customise a Pi image

This is most likely the bit you've done already, Pick an image from the Pi downloads section install it and configure it the way you like. This might include:

- updateing packages
- autostarting graphical mode
- adding specific software
- branding for your instituion
- remove unwanted stuff

NB. When I first did this I found that the filesysytem on my custom image was corrupt and I had to redo the entire job. Ensure you shut down the Pi carefully when setting up your image.

```
sudo shutdown -h now
```

STEP 2 – Create an initial backup image using dd

Now that your Pi is configured let's create a backup image. Using you second machine plug in the SD card you just made, wait a moment then run this command to find out how your system refers to the card you just plugged in.

```
sudo fdisk -l
```

This will list all the drives connected to you machine, find the entry similar to this one:

```
Disk /dev/sdc: 3965 MB, 3965190144 bytes
4 heads, 32 sectors/track, 60504 cylinders, total 7744512 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000f01ca

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1            8192      122879       57344    6  FAT16
/dev/sdc2          122880     6877183     3377152   83  Linux
/dev/sdc3         6877184     7905279      514048    b  W95 FAT32
```

In my case the SD has been recognised as /dev/sdc. We can backup our card using the dd command below.

```
sudo dd bs=4M if=/dev/sdc of=~/Pi_backup.img
```

This will create a backup of the SD card called “Pi_backup.img”.

STEP 3 – Mounting backed up image.

We can treat this file like a connected device using special loopback mount point /dev/loop0. We will use the *losetup* command and *kpartx* to mount the device and find the partitions. You will likely need to install the *kpartx* command. On Debian / Ubuntu type

```
sudo apt-get install kpartx
```

When installed type

```
sudo losetup /dev/loop0 Pi_backup.img
```

This attaches the image file as a device. Then we need to identify the partitions within the image:

```
kpartx -av /dev/loop0
```

Prior to mounting we need to create some mount points for each partition in the old and new images.

```
mkdir -p /tmp/Pi_img/old1 /tmp/Pi_img/old2 /tmp/Pi_img/new1 /tmp/Pi_img/new2
```

Then mount the two partitions from the existing image.

```
sudo  mount /dev/mapper/loop0p1 /tmp/Pi_img/old1
sudo  mount /dev/mapper/loop0p2 /tmp/Pi_img/old2
```

Lastly we need to find out the size of the files on each partition by typing

```
df
```

Check the output and add together the used space for the 2 mounted partitions and add an extra 200000. This will be the overall size of the the new image we will create in the next step.

```
Filesystem          1K-blocks      Used Available Use% Mounted on
/dev/mapper/loop0p1     57288     18888     38400  33% /tmp/Noobs-Converter/Orig/1
/dev/mapper/loop0p2   3716304   2049184   1496444  58% /tmp/Noobs-Converter/Orig/2
```

In my case I add 18888, 2049184 and 200000 which gives my a size of **2268072**.

Step 4 - Creating and partitioning a new empty image

We now need to create a Noobs compatible image which we can copy our data into, first create an empty image file.

```
dd if=/dev/zero of=/Pi_Backup-Noobs bs=1024 count=0 seek=2268072
```

Next we will attach this image as loopback device loop1

```
sudo losetup /dev/loop1 Pi_Backup-Noobs
sudo kpartx -av /dev/loop1
```

Now we create a new partition table and the logical partition needed by Noobs

```
sudo parted -s /dev/loop1 mklabel msdos
sudo parted -s /dev/loop1 mkpart primary fat16 4194KB 62.9MB
sudo parted -s /dev/loop1 mkpart extended 62.9MB 100%
sudo parted -s /dev/loop1 mkpart logical 62.9MB 100%
sudo kparrtx -av /dev/loop1
```

Next we format the partitions (NB the new main partition we've made is p5 as it is a logical partition)

```
sudo mkfs.vfat /dev/mapper/loop1p1
sudo mkfs.ext4 /dev/mapper/loop1p5
```

Lastly we can mount these partitions

```
sudo mount /dev/mapper/loop1p1 /tmp/Pi_img/new1
sudo mount /dev/mapper/loop1p5 /tmp/Pi_img/new2
```

Step 5 – Copying Data & tidying up

We can now transfer data from the old image to the new one

```
sudo cp -Rp /tmp/Pi_img/old1/* /tmp/Pi_img/new/1/
sudo cp -Rp /tmp/Pi_img/old2/* /tmp/Pi_img/new/2/
```

This may take a little while to copy. We then need to make a really important tweak to cmdline.txt found in the first partition. We need to replace the *root=/dev/mmcblk0p2* with *root=/dev/mmcblk0p5,* this can be done manually or with the command.

```
sudo sed -i 's/mmcblk0p2/mmcblk0p5/g' /tmp/Noobs-Converter/New/1/cmdline.txt
```

Once this is done we can umount and disconnect our loopback devices.

```
sudo umount /dev/mapper/loop0p1 /dev/mapper/loop0p2 /dev/mapper/loop1p1 /dev/mapper/loop1p5
kpartx -d /dev/loop1
kpartx -d /dev/loop0
losetup -d /dev/loop0
losetup -d /dev/loop1
```

Lastly we remove our /tmp data

```
rm -rf /tmp/Pi_img
```

Step 6 – Compress the new image and add to the Noobs installer.

You can compress the image in other ways, however the official Noobs images and compressed using xz, so we'll follow suit.

```
xz -e -9 Pi_Backup-Noobs.img
```

Now this will take a long time, but once complete we're done. This image should now be compatible with Noobs.

All that's left is to download Noobs and add your image to those in the Images folder before writing to an SD card.