

welcome: [please sign in](#)

## Raspbian FAQ

### Contents

1. [Raspbian FAQ](#)
  1. [I'm new to Raspberry Pi and Raspbian. What should I know?](#)
  2. [What is Raspbian?](#)
  3. [What do you mean by "soft float ABI" and "hard float ABI"?](#)
  4. [What is Debian?](#)
  5. [Is there documentation available for Raspbian?](#)
  6. [How much of Debian wheezy armhf does Raspbian support?](#)
  7. [Will Raspbian work with non-Raspberry Pi systems?](#)
  8. [Are you planning to have an official SD image for the Raspberry Pi?](#)
  9. [How can I use Raspbian packages until the official SD image is ready?](#)
  10. [What will be on the official Raspbian SD image?](#)
  11. [Is the Raspbian repository live? If so, where can I find it?](#)
  12. [Do you have a list of all the packages in the Raspbian repository?](#)
  13. [What do I need in my sources.list file to access the Raspbian repository?](#)
  14. [Can I mix packages from the Debian repositories with Raspbian?](#)
  15. [Are 'contrib' and/or 'non-free' included the repository?](#)
  16. [Is it just the repository? Or, will you provide any customization?](#)
  17. [What compilation options should be set Raspbian code?](#)
2. [How do I ...?](#)
  1. [How do I mount a directory from a Windows server? Where's smbmount and smbfs?](#)
  2. [How do I enable or use IPv6?](#)
  3. [How do I mount directories from other Linux/Unix/BSD servers? How do I mount an NFS share?](#)
  4. [How do I make a proper root account?](#)
  5. [No space left on device, but I have](#)

plenty of free space on my SD card.  
What do I do?

## I'm new to Raspberry Pi and Raspbian. What should I know?

---

The goal of Raspbian is to become the leading OS of choice for all users of the Raspberry Pi. This goal has been largely achieved. There are still some rough edges but these are generally more related to the Pi's hardware and kernel than Raspbian itself.

Raspbian images are produced by various people. For newcomers we recommend the images provided by the Raspberry Pi foundation, these are available from their download page [Raspberry Pi Foundation's download page](#).

Raspbian tries to stay as close to Debian as reasonably possible. Debian is used by millions of users around the globe on a daily basis and there is a vast store of knowledge and documentation about using Debian across the web. Any information you find that applies to Debian will almost certainly apply to the same version of Raspbian. Information for earlier versions of debian will often apply too but may need some tweaks just as it would when using it on a newer version of debian.

## What is Raspbian?

---

Raspbian is an unofficial port of Debian wheezy armhf with compilation settings adjusted to produce code that uses "hardware floating point", the "hard float" ABI and will run on the Raspberry Pi.

The port is necessary because the official Debian wheezy armhf release is compatible only with versions of the ARM architecture later than the one used on the Raspberry Pi (ARMv7-A CPUs and higher, vs the Raspberry Pi's ARMv6 CPU).

The Debian squeeze image issued by the Raspberry Pi foundation was based on debian armel which uses software floating point and the "soft float" ABI. The foundation used the existing Debian port for less capable ARM devices. Therefore, it did not use of the Pi's processor's floating point hardware - reducing the Pi's performance during floating point intensive applications - or the advanced instructions of the ARMv6 CPU.

## What do you mean by "soft float ABI" and "hard float ABI"?

---

ABI stands for application binary interface. The term has slightly different meanings in different contexts but in this context it refers to the set of rules used to set up registers and the stack when calling and returning from functions and other details that are required for binaries to be compatible with each other.

The "soft float" ABI passes floating point parameters in integer registers while the

"hard float ABI" passes them in vfp (floating point) registers. The two ABIs are not compatible because they use different registers. It is possible to use hardware floating point with the soft-float ABI but doing so means that whenever a floating point value is passed to or returned from a function it must be transferred to an integer register incurring a performance penalty.

## What is Debian?

---

Debian is a free operating system for your computer and includes the basic set of programs and utilities that make your computer run along with many thousands of other packages. Debian has a reputation within the Linux community for being very high-quality, stable and scalable. Debian also has an extensive and friendly user community that can help new users with support for practically any problem. This makes Debian an ideal operating system for the Raspberry Pi that will be used by children and many others using Linux for the first time.

Further information about what Debian is can be found [here](#), [here](#), and [here](#).

## Is there documentation available for Raspbian?

---

Because Raspbian is closely based on Debian, there is an enormous amount of documentation available for Raspbian. Aside from the [Debian.org](#) web site and the wealth of information there, a terrific source of documentation is the recently released [The Debian Administrator's Handbook](#) which is available in hard-copy or as a free download. Note however that the current version of that handbook is based on debian squeeze while the first release of Raspbian is based on Debian wheezy. Therefore there may be differences in some areas.

Any slight differences between Debian and Raspbian or aspects of Raspbian that are unique to the Raspberry Pi hardware will be documented on this web site or on the Raspberry Pi section of the [elinux.org](#) wiki.

## How much of Debian wheezy armhf does Raspbian support?

---

We have built the vast majority of packages from debian wheezy, indeed there are now more binary package in Raspbian wheezy than there are in the armel and armhf ports of Raspbian (though part of this is that we haven't been as aggressive about removals as debian is).

Please feel free to get in touch through the forums if a package you require is missing. We can't guarantee to provide it but we should at least be able to explain why it is not available and may be able to suggest alternatives.

Note that we don't currently build binaries for contrib or non-free (though we do import architecture independent packages from those areas of the debian archive)

## Will Raspbian work with non-Raspberry Pi systems?

---

Yes. The packages are being compiled for the ARMv6+VFP CPU. If the hardware supports that CPU feature set then Raspbian should function well on that hardware. A different kernel will of course be needed and it should be noted that some hardware has poor support for "regular" Linux system. An example of such hardware is the [Via APC](#). That being said, many newer hardware devices utilize a variant of the ARMv7 CPU which may be better served by the official Debian wheezy armhf port.

## Are you planning to have an official SD image for the Raspberry Pi?

---

There are no immediate plans for Raspbian itself to release official SD card images. Currently we recommend beginners use the images provided by the Raspberry Pi foundation. Some other images are listed on the [Images page](#).

## How can I use Raspbian packages until the official SD image is ready?

---

You can try unofficial SD images listed on the [Images page](#) from the such as those from [Hexxeh](#) or the [Pisces Images](#) from project leader Mike Thompson. Alternatively you can install Raspbian using traditional Debian installation techniques involving the [debootstrap utility](#). Raspbian packages can also be installed in a Debian chroot environment on a Debian armel or armhf system capable of running Raspbian ARMv6 code. This is how the Raspbian port is being bootstrapped.

## What will be on the official Raspbian SD image?

---

At this time it is impossible to determine what exactly will be provided on the official Raspbian SD image. However, at a minimum it will include a basic Debian wheezy install with development tools and a GUI environment. In most respects, the initial SD image will probably look very similar to the [Debian Squeeze armel SD image](#) released earlier by the Raspberry Pi foundation. Except of course, it will be compiled for optimal execution speed on the ARMv6 CPU of the Raspberry Pi.

## Is the Raspbian repository live? If so, where can I find it?

---

Yes, the Raspbian repository is live and is growing in size by a few hundred packages a day. It can be found at the following link:

<http://archive.raspbian.org/>

## Do you have a list of all the packages in the

## Raspbian repository?

---

The current list of packages in the Raspbian repository can be found in the text file linked below:

<http://archive.raspbian.org/raspbian/dists/wheezy/main/binary-armhf/Packages>

## What do I need in my sources.list file to access the Raspbian repository?

---

Your **/etc/apt/sources.list** file should look as follows:

```
deb http://archive.raspbian.org/raspbian wheezy main contrib  
non-free rpi  
deb-src http://archive.raspbian.org/raspbian wheezy main contrib  
non-free rpi
```

When using apt-get, you will want to install the Raspbian public key in your apt-get keyring, it should already be there if you used debootstrap from the Raspbian repository but if you need to add it manually this can be done with following command:

```
wget http://archive.raspbian.org/raspbian.public.key -O - | sudo  
apt-key add -
```

Mirrors of Raspbian that may be more suitable for users in different geographic locations can be found [here](#).

## Can I mix packages from the Debian repositories with Raspbian?

---

The short answer is if you don't know what you are doing you should not mix them. For the long answer read on.

Debian armel packages use the soft float ABI which is incompatible with the hard float ABI used by Raspbian. In theory it should be possible to install Debian armel packages in parallel with Raspbian packages using [multiarch](#). However multiarch setups conflict with a hack we had to make to support the videocore libraries and there are other potential issues too. As such we don't currently recommend or support multiarch configurations with Debian armel and Raspbian armhf.

Debian armhf packages should be compatible with Raspbian packages but a system with such a mixture WILL NOT RUN ON THE PI. Furthermore there may be corner cases where libraries build slightly differently on Raspbian. Such mixed systems can be useful for development (they were used heavily in the process of creating Raspbian) but are not recommended for general use.

Architecture independent ("arch all") packages are compatible between Debian and

Raspbian. Source packages should in general be compatible though some may need modification to adjust compiler settings (most Debian packages just use the compiler defaults but some use their own settings for various reasons). A test script for testing packages for armv7 contamination is available at <http://pastebin.com/BtSdvrXM> but the script is not perfect and can suffer from both false positives and false negatives.

## **Are 'contrib' and/or 'non-free' included the repository?**

---

Yes. However, at this point, we have only brought over the source code and the architecture 'all' packages. This should allow many USB devices to be supported as their firmware is often found in architecture 'all' packages. In the future, we'll be enabling compilation of the architecture specific 'armhf' packages, but this requires further tweaking of our autobuild infrastructure. Currently it is not planned to build contrib and non-free until we migrate to new build infrastructure after the release of wheezy.

## **Is it just the repository? Or, will you provide any customization?**

---

In general, the Raspbian repository will be kept as close to Debian wheezy as possible with changes limited to those we deem essential so that a given package functions on Raspberry Pi hardware. This approach will appeal to fans of Debian who simply want the optimal Debian experience on Raspberry Pi hardware.

An area in the repository called "rpi" was created for Raspberry Pi specific packages but none have been placed in there yet due to a combination of lack of manpower and the need to coordinate with the raspberry Pi foundation. The raspberry Pi foundation have their own repository at <http://archive.raspberrypi.org/debian/> which contains various packages they include in their images.

There is also a repository at <http://archive.raspbian.org/mate> which contains a raspbian build of the [mate desktop environment](#) (a fork of gnome 2).

## **What compilation options should be set Raspbian code?**

---

The compiler tools included in the Raspbian repository are, by default, pre-configured to produce code compatible with Raspbian. However, if you are looking to port compiler tools over to Raspbian, the settings required for most GNU tools are as follows:

```
-march=armv6  
-mfpu=vfp  
-mfloat-abi=hard
```

These settings produce code with armv6 specific instructions, vector floating point

instructions and specify the ARM EABI where floating point values are to be passed in floating point registers. Additional information can be found in the [Debian hard float documents](#).

## How do I ...?

### How do I mount a directory from a Windows server? Where's smbmount and smbfs?

---

*smbfs* has been [replaced in Debian](#) (and Raspbian) by *cifs*. The utility *mount.cifs* is in the package [cifs-utils](#).

Also, the kernel you are using must be built to support mounting SMB/CIFS volumes.

### How do I enable or use IPv6?

---

Easy. *sudo modprobe ipv6* will enable it immediately. If you want it to automatically be enabled at boot, just add *ipv6* on a line by itself at the end of */etc/modules*.

### How do I mount directories from other Linux/Unix/BSD servers? How do I mount an NFS share?

---

First, you need a server that can use the Network File System (NFS). Let's assume you have a server named *adelaide* at 192.168.222.34 and it has a share called *public* that you want to mount at */public* on your Pi.

First, install the necessary packages. I use *aptitude*, but you can use *apt-get* instead if you prefer; just put *apt-get* where you see *aptitude*:

```
sudo aptitude install nfs-common portmap
```

(*nfs-common* may already be installed, but this will ensure that it installs if it isn't already)

On the current version of Raspbian, *rpcbind* (part of the *portmap* package) does not start by default. This is presumably done to control memory consumption on our small systems. However, it isn't very big and we need it to make an NFS mount. To enable it manually, so we can mount our directory immediately:

```
sudo service rpcbind start
```

To make *rpcbind* start automatically at boot:

```
sudo update-rc.d rpcbind enable
```

Now, let's mount our share:

Make a directory for your mount. I want mine at */public*, so:

```
mkdir /public
```

Mount manually to test (remember, my server is adelaide at 192.168.222.34 and the share is *public*):

```
mount -t nfs 192.168.222.34:/public /public
```

(my server share path) (where I want it mounted locally)

Now, to make it permanent, you need to edit `/etc/fstab` to make the directory mount at boot. I added a line to the end of `/etc/fstab`:

```
192.168.222.34:/public /public nfs rsize=8192,wsiz=8192,timeo=14,intr 0 0
```

The `rsize` and `wsiz` entries are not strictly necessary but I find they give me some extra speed.

There are ways you can refer to your server by name instead of by its IP address like we did above. Have a look at `/etc/hosts` for a starting point. The rest is left as an exercise for the reader.

## How do I make a proper root account?

---

Don't do this unless you're certain that you want and need a true root account. `sudo` will take care of root tasks for you nicely in almost all cases.

However, if you do want a root account so you can log in as root or `su` to a root account, just type:

```
sudo passwd
```

This will let you enter a password for the root account. Be careful to make it a secure one!

One other modification you should do, if you enable a root account, is to forbid root logins via `ssh` for security reasons. You can log in using a normal user account instead (such as `pi`), and use the `su` command to become root. To do this:

```
cd /etc/ssh
```

```
sudo nano sshd_config
```

 (you can do this as root too of course; just don't type `sudo` in front of the command)

Find the line:

```
PermitRootLogin yes
```

Change `yes` to `no`. (If yours already says `no` then there is nothing to do.)

Save the config file, and to make it take effect:

```
sudo service ssh restart
```



## No space left on device, but I have plenty of free space on my SD card. What do I do?

---

The first thing to check is that the partition actually fills the SD card. Most images are built with partitions sized to fit on relatively small cards. The Raspberry Pi foundation image contains a tool called raspi-config which contains an option to enlarge the root partition. Other setups may require other procedures to resize the partitions which are beyond the scope of this FAQ.

If there is plenty of space on the root partition then it may be a /tmp issue. On many Raspbian systems, for performance reasons, the /tmp directory is a RAM disk with limited memory. Installers for applications will sometimes fill up this temporary space and you may see messages such as the one below.

```
cp: writing `/tmp/tmp.XwAcjvnTkE
/rfc5834.txt.gz': No space left on device
```

A way of working around this is to edit the file:

*/etc/default/rcS*

change the last line from

*RAMTMP=yes*

to

*RAMTMP=no*

and then reboot.

Then try reinstalling your package.

Change the line back to *RAMTMP=yes* when you have finished.