

Raspberry Pi OS

Introduction

Edit this [on GitHub](#)

Raspberry Pi OS is a free operating system based on Debian, optimised for the Raspberry Pi hardware, and is the recommended operating system for normal use on a Raspberry Pi. The OS comes with over 35,000 packages: pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.

Raspberry Pi OS is under active development, with an emphasis on improving the stability and performance of as many Debian packages as possible on Raspberry Pi.

Updating and Upgrading Raspberry Pi OS

Edit this [on GitHub](#)

It's important to keep your Raspberry Pi up to date. The first and probably the most important reason is security. A device running Raspberry Pi OS contains millions of lines of code that you rely on. Over time, these millions of lines of code will expose well-known vulnerabilities, which are documented in [publicly available databases](#) meaning that they are easy to exploit. The only way to mitigate these exploits as a user of Raspberry Pi OS is to keep your software up to date, as the upstream repositories track CVEs closely and try to mitigate them quickly.

The second reason, related to the first, is that the software you are running on your device most certainly contains bugs. Some bugs are CVEs, but bugs could also be affecting the desired functionality without being related to security. By keeping your software up to date, you are lowering the chances of hitting these bugs.

Using APT

The easiest way to manage installing, upgrading, and removing software is using APT (Advanced Packaging Tool) from Debian. To update software in Raspberry Pi OS, you can use the `apt` tool from a Terminal window.

APT keeps a list of software sources on your Raspberry Pi in a file at `/etc/apt/sources.list`. Before installing software, you should update your package list with `apt update`. Go ahead and open a Terminal window and type:

```
sudo apt update
```

Next, **upgrade** all your installed packages to their latest versions with the following command:

```
sudo apt full-upgrade
```

Note that `full-upgrade` is used in preference to a simple `upgrade`, as it also picks up any dependency changes that may have been made.

Generally speaking, doing this regularly will keep your installation up to date for the particular major Raspberry Pi OS release you are using (e.g. Buster). It will not update from one major release to another, for example, Stretch to Buster or Buster to Bullseye.

However, there are occasional changes made in the Raspberry Pi OS image that require manual intervention, for example a newly introduced package. These are not installed with an upgrade, as this command only updates the packages you already have installed.

NOTE

The kernel and firmware are installed as a Debian package, and so will also get updates when using the procedure above. These packages are updated infrequently and after extensive testing.

If moving an existing SD card to a new Raspberry Pi model (for example the Raspberry Pi Zero 2 W), you may also need to update the kernel and the firmware first using the instructions above.

Running Out of Space

When running `sudo apt full-upgrade`, it will show how much data will be downloaded and how much space it will take up on the SD card. It's worth checking with `df -h` that you have enough free disk space, as

unfortunately `apt` will not do this for you. Also be aware that downloaded package files (`.deb` files) are kept in `/var/cache/apt/archives`. You can remove these in order to free up space with `sudo apt clean` (`sudo apt-get clean` in older releases of `apt`).

Upgrading from Previous Operating System Versions

WARNING

Upgrading an existing image is possible, but is not guaranteed to work in every circumstance and we do not recommend it. If you do wish to try upgrading your operating system version, we strongly suggest making a backup first — we can accept no responsibility for loss of data from a failed update.

The latest version of Raspberry Pi OS is based on [Debian Bullseye](#). The previous version was based on [Buster](#). If you want to perform an in-place upgrade from Buster to Bullseye (and you're aware of the risks) see the [instructions in the forums](#).

Searching for Software

You can search the archives for a package with a given keyword with `apt-cache search`:

```
apt-cache search locomotive
sl - Correct you if you type `sl' by mistake
```

You can view more information about a package before installing it with `apt-cache show`:

```
apt-cache show sl
Package: sl
Version: 3.03-17
Architecture: armhf
Maintainer: Hiroyuki Yamamoto <yama1066@gmail.com>
Installed-Size: 114
Depends: libc6 (>= 2.4), libncurses5 (>= 5.5-5~), libtinfo5
Homepage: http://www.tkl.iis.u-tokyo.ac.jp/~toyoda/index_e.html
Priority: optional
Section: games
Filename: pool/main/s/sl/sl_3.03-17_armhf.deb
Size: 26246
SHA256: 42dea9d7c618af8fe9f3c810b3d551102832bf217a5bcdba310f119f62117dfb
SHA1: b08039acccecd721fc3e6faf264fe59e56118e74
MD5sum: 450b21cc998dc9026313f72b4bd9807b
Description: Correct you if you type `sl' by mistake
 Sl is a program that can display animations aimed to correct you
 if you type 'sl' by mistake.
 SL stands for Steam Locomotive.
```

Installing a Package with APT

```
sudo apt install tree
```

Typing this command should inform the user how much disk space the package will take up and asks for confirmation of the package installation. Entering `Y` (or just pressing `Enter`, as yes is the default action) will allow the installation to occur. This can be bypassed by adding the `-y` flag to the command:

```
sudo apt install tree -y
```

Installing this package makes `tree` available for the user.

Uninstalling a Package with APT

You can uninstall a package with `apt remove`:

```
sudo apt remove tree
```

The user is prompted to confirm the removal. Again, the `-y` flag will auto-confirm.

You can also choose to completely remove the package and its associated configuration files with `apt purge`:

```
sudo apt purge tree
```

Using `rpi-update`

`rpi-update` is a command line application that will update your Raspberry Pi OS kernel and VideoCore firmware to the latest pre-release versions.

WARNING

Pre-release versions of software are not guaranteed to work. You should not use `rpi-update` on any system unless recommended to do so by a Raspberry Pi engineer. It may leave your system unreliable or even completely broken. It should not be used as part of any regular update process.

The `rpi-update` script was originally written by [Hexxeh](#), but is now supported by Raspberry Pi engineers. The script source is in the [rpi-update repository](#).

What it does

`rpi-update` will download the latest pre-release version of the linux kernel, its matching modules, device tree files, along with the latest versions of the VideoCore firmware. It will then install these files to relevant locations on the SD card, overwriting any previous versions.

All the source data used by `rpi-update` comes from the [rpi-firmware repository](#). This repository simply contains a subset of the data from the [official firmware repository](#), as not all the data from that repo is required.

Running `rpi-update`

If you are sure that you need to use `rpi-update`, it is advisable to take a backup of your current system first as running `rpi-update` could result in a non-booting system.

`rpi-update` needs to be run as root. Once the update is complete you will need to reboot.

```
sudo rpi-update
sudo reboot
```

It has a number of options documented in the [rpi-update repository](#).

How to get back to safety

If you have done an `rpi-update` and things are not working as you wish, if your Raspberry Pi is still bootable you can return to the stable release using:

```
sudo apt-get update
sudo apt install --reinstall libraspberrypi0 libraspberrypi-{bin,dev,doc}
raspberrypi-bootloader raspberrypi-kernel
```

You will need to reboot your Raspberry Pi for these changes to take effect.

Playing Audio and Video

Edit this [on GitHub](#)

WARNING

The following documentation refers to Raspberry Pi OS Buster and earlier versions. OMXPlayer has been deprecated in the [latest OS release](#). If you are running Bullseye, VLC is now the recommended alternative.

The simplest way of playing audio and video on Raspberry Pi is to use the installed OMXPlayer application.

This is hardware accelerated, and can play back many popular audio and video file formats. OMXPlayer uses the OpenMAX (`omx`) hardware acceleration interface (API) which is the officially supported media API on Raspberry Pi. OMXPlayer was developed by the Kodi Project's Edgar Hucek.

The OMXPlayer Application

The simplest command line is `omxplayer <name of media file>`. The media file can be audio or video or both. For the examples below, we used an H264 video file that is included with the standard Raspberry Pi OS installation.

```
omxplayer /opt/vc/src/hello_pi/hello_video/test.h264
```

By default the audio is sent to the analog port. If you are using a HDMI-equipped display device with speakers, you need to tell omxplayer to send the audio signal over the HDMI link.

```
omxplayer --adev hdmi /opt/vc/src/hello_pi/hello_video/test.h264
```

When displaying video, the whole display will be used as output. You can specify which part of the display you want the video to be on using the window option.

```
omxplayer --win 0,0,640,480 /opt/vc/src/hello_pi/hello_video/test.h264
```

You can also specify which part of the video you want to be displayed: this is called a crop window. This portion of the video will be scaled up to match the display, unless you also use the window option.

```
omxplayer --crop 100,100,300,300 /opt/vc/src/hello_pi/hello_video/test.h264
```

If you are using the [Raspberry Pi Touch Display](#), and you want to use it for video output, use the display option to specify which display to use. `n` is 5 for HDMI, 4 for the touchscreen. With the Raspberry Pi 4 you have two options for HDMI output. `n` is 2 for HDMI0 and 7 for HDMI1.

```
omxplayer --display n /opt/vc/src/hello_pi/hello_video/test.h264
```

How to Play Audio

To play an MP3 file, navigate to the location of the `.mp3` file in the terminal using `cd` and then type the following command:

```
omxplayer example.mp3
```

This will play the audio file `example.mp3` through either your monitor's built-in speakers or your headphones, connected via the headphone jack.

If you need an example file you can download one from here using the following command:

```
wget https://raw.githubusercontent.com/raspberrypilearning/burping-jelly-baby/master/data/la.mp3 -O example.mp3 --no-check-certificate
```

If you cannot hear anything, make sure your headphones or speakers are connected correctly. Note that omxplayer doesn't use ALSA and so ignores the [audio configuration](#) set by `raspi-config` or `amixer`.

If omxplayer's auto-detection of the correct audio output device fails, you can force output over HDMI with:

```
omxplayer -o hdmi example.mp3
```

Alternatively, you can force output over the headphone jack with:

```
omxplayer -o local example.mp3
```

You can even force output over both the headphone jack and HDMI with:

```
omxplayer -o both example.mp3
```

How to Play Video

To play a video, navigate to the location of your video file in the terminal using `cd`, then type the following command:

```
omxplayer example.mp4
```

This will play the `example.mp4` in full screen. Hit `Ctrl + C` to exit.

On the Raspberry Pi 4, hardware support for MPEG2 and VC-1 codecs has been removed, so we recommend the use of the VLC application, which supports these formats in software. In addition, VLC has hardware support for H264 and the new HEVC codec.

An Example Video

A video sample of the animated film *Big Buck Bunny* is available on your Raspberry Pi. To play it enter the following command into a terminal window:

```
omxplayer /opt/vc/src/hello_pi/hello_video/test.h264
```

On a Raspberry Pi 4, use the following command for H264 files:

```
omxplayer /opt/vc/src/hello_pi/hello_video/test.h264
```

or for H264, VC1, or MPEG2

```
vlc /opt/vc/src/hello_pi/hello_video/test.h264
```

When using VLC, you can improve playback performance by encapsulating the raw H264 stream, for example from the Raspberry Pi Camera Module. This is easily done using `ffmpeg`. Playback is also improved if VLC is run full screen;

either select fullscreen from the user interface, or you can add the `--fullscreen` options to the `vlc` command line.

This example command converts `video.h264` to a containerised `video.mp4` at 30 fps:

```
ffmpeg -r 30 -i video.h264 -c:v copy video.mp4
```

Options During Playback

There are a number of options available during playback, actioned by pressing the appropriate key. Not all options will be available on all files. The list of key bindings can be displayed using `omxplayer --keys`:

```
1          decrease speed
2          increase speed
<          rewind
>          fast forward
z          show info
j          previous audio stream
k          next audio stream
i          previous chapter
o          next chapter
n          previous subtitle stream
m          next subtitle stream
s          toggle subtitles
w          show subtitles
x          hide subtitles
d          decrease subtitle delay (- 250 ms)
f          increase subtitle delay (+ 250 ms)
q          exit omxplayer
p / space  pause/resume
-          decrease volume
+ / =      increase volume
left arrow seek -30 seconds
right arrow seek +30 seconds
down arrow seek -600 seconds
up arrow   seek +600 seconds
```

Playing in the Background

`omxplayer` will close immediately if run in the background without tty (user input), so to run successfully, you need to tell `omxplayer` not to require any user input using the `--no-keys` option.

```
omxplayer --no-keys example.mp3 &
```


Adding the `&` at the end of the command runs the job in the background. You can then check the status of this background job using the `jobs` command. By default, the job will complete when `omxplayer` finishes playing, but if necessary, you can stop it at any point using the `kill` command.

```
$ jobs
[1]-  Running                  omxplayer --no-keys example.mp3 &
$ kill %1
$
[1]-  Terminated              omxplayer --no-keys example.mp3 &
```

Using a USB webcam

Edit this [on GitHub](#)

Rather than using the Raspberry Pi [camera module](#), you can use a standard USB webcam to take pictures and video on your Raspberry Pi.

NOTE

The quality and configurability of the camera module is highly superior to a standard USB webcam.

First, install the `fswebcam` package:

```
sudo apt install fswebcam
```

If you are not using the default `pi` user account, you need to add your username to the `video` group, otherwise you will see 'permission denied' errors.

```
sudo usermod -a -G video <username>
```

To check that the user has been added to the group correctly, use the `groups` command.

Basic Usage

Enter the command `fswebcam` followed by a filename and a picture will be taken using the webcam, and saved to the filename specified:

```
fswebcam image.jpg
```

This command will show the following information:

```
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
```

```
Adjusting resolution from 384x288 to 352x288.  
--- Capturing frame...  
Corrupt JPEG data: 2 extraneous bytes before marker 0xd4  
Captured frame in 0.00 seconds.  
--- Processing captured image...  
Writing JPEG image to 'image.jpg'.
```