PipeWire is a server and API for handling multimedia on Linux. Its most common use is for Wayland and Flatpak applications to implement screensharing, remote desktop, and other forms of audio and video routing between different pieces of software. Per the official FAQ, "you can think of it as a multimedia routing layer on top of the drivers that applications and libraries can use."

As opposed to PulseAudio's focus on consumer audio and JACK's focus on professional audio, PipeWire aims to work for all users at all levels. Among other techniques, PipeWire achieves this with its ability to dynamically switch between different buffer sizes, for adapting to the different latency requirements of different audio applications.

In Debian 10, PipeWire 0.2.5 is available, and should not need to manually be installed, as it's usually brought in as a dependency by applications that make use of it.

In Debian 11, PipeWire 0.3.19 is available, and can be experimentally used as a substitute for the ALSA userspace library, PulseAudio, and JACK. This is a documented but unsupported use-case.

In Debian 12, PipeWire 0.3.65 is available, and is considerably more reliable, and is a comfortable drop-in replacement for many use-cases. PipeWire is the default sound server with GNOME Desktop.

Much newer versions are available in Debian Testing, Unstable and in Backports repositories.

Contents

Installation

In its most basic form, PipeWire can be installed simply by installing the DebianPkg: pipewire package.

In Debian 10, some plugins are split into separate packages. These can be found in the DebianPkg: libspa-bluetooth and DebianPkg: libspa-ffmpeg packages.

In Debian 11, there is no longer a separate FFmpeg plugin package, and the Bluetooth package has been moved to DebianPkg: libspa-0.2-bluetooth. There is also a JACK plugin in the DebianPkg: libspa-0.2-jack package, required for using PipeWire as a JACK server (as explained below).

For Debian 11, PipeWire can be used to replace PulseAudio. Additionally, the DebianPkg: pipewire-audio-client-libraries package is available. The libraries in this package allow applications directly accessing JACK and ALSA to output via the PipeWire server, allowing harmonious interoperability between the different sound servers and APIs.

For Debian 12 and newer, DebianPkg: pipewire-audio-client-libraries is replaced by DebianPkg: pipewire-alsa and DebianPkg: pipewire-jack. It is **recommended** to install the metapackage DebianPkg: pipewire-audio which depends on DebianPkg: wireplumber (the recommended session manager), DebianPkg: pipewire-pulse (to replace PulseAudio), DebianPkg: pipewire-alsa (ALSA) and DebianPkg: libspa-0.2-bluetooth (for Bluetooth support). Moreover, installing this metapackage will remove pulseaudio to prevent any conflicts between both sound server.

Configuration

Using as a substitute for PulseAudio/JACK/ALSA

Debian 11

> As per Simon McVittie, "This is not a supported scenario for Debian 11, and is considered experimental."
>
> It may break certain applications, lack certain functionality, or not work at all.

Users may also be interested in PipeWire's capacity to be a drop-in substitute for existing Linux audio solutions, including PulseAudio and JACK. Potential benefits include lower CPU usage, better support for Bluetooth devices, and better integration between applications using JACK and PulseAudio. This is the default behavior in some other distros, notably Fedora 34 and newer.

This behavior can be experimentally enabled in Debian 11.

The three instructional sections below are independent of eachother, but you are still highly recommended to use PipeWire to replace all of them if you intend to replace any of it, for the best integration between different applications.

PulseAudio

In this scenario, PipeWire runs as a per-user audio service, replacing

PulseAudio completely.

Create this empty file:

```
# touch /etc/pipewire/media-session.d/with-pulseaudio
```

Create a `pipewire-pulse` service by copying the example files:

```
# cp /usr/share/doc/pipewire/examples/systemd/user/pipewire-p
```

Run these three commands as your regular user (not as root):

```
// Check for new service files with:
systemctl --user daemon-reload
// Disable and stop the PulseAudio service with:
systemctl --user --now disable pulseaudio.service pulseaudio.
// Enable and start the new pipewire-pulse service with:
systemctl --user --now enable pipewire pipewire-pulse
```

You can check which server is in use by, as your regular user, running:

```
LANG=C pactl info | grep '^Server Name'
```

If it's using PipeWire, the server name will be "PulseAudio (On PipeWire 0.3.19)"

Ensure this continues working after a reboot. If not, you may need to "mask" the PulseAudio service by running:

```
systemctl --user mask pulseaudio
```

This may require another reboot, after which PipeWire should be used exclusively, even following any more reboots. This will block the PulseAudio service from being activated outright. If you wish to restore previous functionality, you'll need to run the same command again with the "unmask" action.

root

If you wish to run PipeWire as root, remove or comment this line:

```
ConditionUser=!root
```

From "pipewire.socket" and "pipewire.service" in "/usr/lib/systemd /user".

And from "pipewire-pulse.service" and "pipewire-pulse.socket" in "/etc/xdg/systemd/user".

ALSA

ALSA clients can be configured to output via PipeWire.

Some may describe this as "replacing ALSA", but as the PipeWire FAQ clarifies: "No, ALSA is an essential part of the Linux audio stack, it provides the interface to the kernel audio drivers. That said, the ALSA user space library has a lot of stuff in it that is probably not desirable anymore these days, like effects plugins, mixing, routing, slaving, etc. PipeWire uses a small subset of the core ALSA functionality to access the hardware. All of the other features should be handled by PipeWire."

Install the DebianPkg: pipewire-audio-client-libraries package, if not already installed.

Create this empty file:

```
# touch /etc/pipewire/media-session.d/with-alsa
```

Copy a configuration file from the PipeWire examples into your ALSA configuration directory:

```
# cp /usr/share/doc/pipewire/examples/alsa.conf.d/99-pipewire
```

JACK

JACK clients can be configured to output via PipeWire instead of JACK.

Install the DebianPkg: pipewire-audio-client-libraries and
 DebianPkg: libspa-0.2-jack packages, if not already installed.

Create this empty file:

```
# touch /etc/pipewire/media-session.d/with-jack
```

Either run JACK clients using the pw-jack wrapper, or copy:

```
# cp /usr/share/doc/pipewire/examples/ld.so.conf.d/pipewire-j
```

And run:

```
# ldconfig
```

This will replace the JACK server libraries with PipeWire's replacements at application runtime, by pointing the dynamic linker at the /usr/lib/x86_64-linux-gnu/pipewire-0.3/jack/ folder.

Bluetooth

PipeWire will generally work fine in conjunction with Bluez to supply high-quality audio to Bluetooth-enabled audio devices when used as a replacement for PulseAudio, with some caveats.

At minimum, you will need to install the DebianPkg: libspa-0.2-bluetooth package, remove the DebianPkg: pulseaudio-module-bluetooth package, and then either reboot your computer or restart the PipeWire services, otherwise device connections will fail with "Protocol not available".

PipeWire will attempt to choose the best possible codec by default. You can override this, and tweak many other related settings, in the `/etc/pipewire/media-session.d/bluez-monitor.conf` file. You can edit this directly, or store local per-user changes by copying the file to `~/.config/pipewire/media-session.d/bluez-monitor.conf` and editing that instead.

Additionally, since the release of PipeWire 0.3.19, there have been numerous Bluetooth-related feature additions and bug fixes. The version in Debian Bullseye is not representative of the current state of PipeWire.

Debian 12

The version of PipeWire present in Debian 12 has vastly improved compatibility and reliability, and is also much easier to configure as a replacement.

Two session managers are now available. The one pulled in by default (DebianPkg: wireplumber) is the one recommended by pipewire's developers. The other one (DebianPkg: pipewire-media-session) is primitive, and is best when using PipeWire just for its basic functionality like screensharing. When using PipeWire as your system's sound server, the maintainer recommends installing the more advanced WirePlumber instead. This command will install WirePlumber while removing the old session manager:

```
# apt install wireplumber pipewire-media-session-
```

You will then need to enable WirePlumber in "systemd" (running as user, not as root):

```
$ systemctl --user --now enable wireplumber.service
```

When using WirePlumber, the amount of required configuration to use PipeWire as your sound server is very minimal. In fact, the WirePlumber package recommends [DebianPkg: pipewire-pulse](), so that will likely happen automatically. You'll only have to enable the ALSA and JACK layers manually, which no longer require creating an empty file in your media session's configuration directory (as that directory won't even exist when using WirePlumber), but will still require copying files from your /usr/share/doc/pipewire/examples/ directory, as described in the Debian 11 instructions.

If you are interested in automatic Bluetooth profile switching (e.g. switching from A2DP to HSP/HFP when an application needs microphone access) WirePlumber gained support for it in release 0.4.8.

A notable change compared to Debian 11 is that the configuration directory has been moved from `/etc/pipewire/` to `/usr/share/pipewire/`: you may want to get rid of the former to avoid confusion.

PulseAudio

Install the [DebianPkg: pipewire-pulse]() package, and reboot. This will configure PipeWire to activate its PulseAudio replacement daemon. Verify that it's enabled by running:

```
LANG=C pactl info | grep '^Server Name'
```

If PipeWire is configured properly, this will print "Server Name: PulseAudio (on PipeWire 0.3.XX)" If it's not working properly, see the Debian 11 instructions for masking PulseAudio.

ALSA

Install the DebianPkg: pipewire-alsa package, and reboot. This will configure PipeWire to activate its !ALSA plugin. Verify that it's enabled by running:

```
LANG=C aplay -L | grep -A 1 default
```

If PipeWire is configured properly, this will print "Default ALSA Output (currently PipeWire Media Server)" If you still see "Playback/recording through the PulseAudio sound server" then Pulseaudio still has control of ALSA output and needs to be uninstalled.

JACK

The Debian 11 instructions still apply, at time of writing.

Keep in mind that the "touch" commands are unnecessary if you're using WirePlumber instead of pipewire-media-session. Refer to the top of this section for more details on which session manager you should be using.

If you're using pipewire-media-session, note that its configuration directory has moved from `/etc/pipewire/media-session.d/` to `/usr/share/pipewire/media-session.d`, and adjust the paths as necessary when making use of the older instructions.

Bluetooth

See the Debian 11 information.

Debian Testing/Unstable

As of August 2023, the pipewire setup for Debian Testing/Unstable is similar to that for Debian 12.

The former session manager DebianPkg: pipewire-media-session won't be shipped with Trixie since it is dead upstream (see ~~Closed in 0.4.2-3+rm: #1030765: pipewire-media-session is dead upstream: 1030765~~).

As advised upstream, if you want to cleanly restart pipewire services, the recommended command is

```
systemctl --user restart wireplumber pipewire pipewire-pulse
```

Checking the output of

```
pw-dump
```

may also help in analysis.

Routing sinks, sources, and audio devices

If you're interested in configuring the plumbing of your sinks and sources with PipeWire, as you usually would with PulseAudio, the easiest way to do this is with either of the tools helvum (GTK based) or qpwgraph (Qt based).

First, ensure that you've configured PipeWire to replace ALSA, JACK, *and* PulseAudio. Then, install either of the DebianPkg: helvum or DebianPkg: qpwgraph packages, and open the installed software.

This will show a graphical representation of every audio device on your system that is acting as an input or output for audio. Additionally, it will display every active program that is producing or receiving audio, and

the connections between all of these.

You can trivially drag to create new connections between applications, audio hardware, or combinations thereof. Some popular uses would be looping your microphone input into your audio output so you can monitor yourself, playing audio through multiple audio output devices at the same time, or using your music player as a fake microphone for other applications receiving voice input. The sky's the limit!

This is only possible with PipeWire, which brings together JACK, PulseAudio and ALSA in a unified graph, allowing arbitrary configuration of programs that may all be using different APIs.

The same task can be accomplished with the PipeWire 🌐 [pw-link](#) command-line tool, which allows creating and destroying links between PipeWire ports (i.e. it replaces JACK's `jack_connect` and `jack_disconnect` tools). Other JACK based tools that can manipulate the graph - like Patchage, Catia, ?RaySession, and the graph window of ?QjackCtl - should also work.

Known issues

pacmd fails with "No PulseAudio daemon running"

The `pacmd` command fails with:

```
$ pacmd
No PulseAudio daemon running, or not running as session daemo
```

The `pactl` command should be roughly equivalent in terms of functionality and does work with a pipewire backend.

skipping and "client missed 1 wakeups"

At least one user ([TheAnarcat](#)) experienced weird "audio drop" and "skipping" issues while running (borg) backups on a workstation under PipeWire in Debian bullseye. The `client missed 1 wakeups` message

showed up in logs and the situation degenerated up to the point where it was not possible to play audio at all. Restarting pipewire fixed the problem:

```
systemctl --user restart pipewire.service
```

This problem doesn't seem to happen anymore in Debian bookworm.

choppy audio on systems with high load

PipeWire's lower latency compared to [PulseAudio](#) can lead to choppy audio on systems with high load.

First view the quantum using the `pw-top` command and then increase the quantum value using this command, until the audio becomes smoother.

```
pw-metadata -n settings 0 clock.force-quantum 2048
```

Once you find the right quantum value for your situation, you can make the value permanent by creating a config file `~/.config/pipewire/pipewire.conf.d/choppy-under-load.conf` with the following content and restart pipewire related daemons.

```
context.properties = {
    default.clock.quantum = 2048
    default.clock.min-quantum = 2048
}
```

See also

- 🌐 [PipeWire homepage](#)
- 🌐 [PipeWire FAQ](#)

- 🌐 [PipeWire Wiki](#)
- 🌐 [PipeWire & WirePlumber Troubleshooting](#)
- [DebianBug:](#) [Debian PipeWire bugs](#)
- 🌐 [Debian security tracker for the PipeWire package](#)
- 🌐 [The state of PipeWire in Summer 2020 (On GNOME Blogs)](#)
- 🌐 [PipeWire 1.0 – An interview with PipeWire creator Wim Taymans (On Fedora Magazine)](#)
- 🌐 [Puppet module deploying the above configuration (except ALSA)](#)

Read the official Debian package documentation for using PipeWire as a replacement for other audio servers by viewing the `/usr/share/doc/pipewire-audio-client-libraries/README.Debian` file.

[CategoryHardware](#) | [CategorySound](#) | [CategorySoftware](#)