

# Dav1d

Christian Külker

2024-05-30

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Links</b>	<b>2</b>
<b>3</b>	<b>Installation</b>	<b>2</b>
3.1	Ansible	3
<b>4</b>	<b>Configuration</b>	<b>3</b>
4.1	Configuration of dav1d and libdav1d6	3
<b>5</b>	<b>Usage</b>	<b>4</b>
<b>6</b>	<b>Compiling Dav1d from Source</b>	<b>6</b>
6.1	Prerequisites	6
6.2	Cloning the Repository	8
6.3	Configuring the Build	8
6.4	Building the Project	8
6.5	Installing	8
6.6	Verification	9
<b>7</b>	<b>History</b>	<b>9</b>
<b>8</b>	<b>Disclaimer of Warranty</b>	<b>9</b>
<b>9</b>	<b>Limitation of Liability</b>	<b>9</b>

## 1 Introduction

This guide provides a detailed overview of dav1d and libdav1d6, a leading open-source cross-platform decoder for AOMedia Video 1 (AV1) video streams focused on speed and

correctness.

Dav1d offers optimized performance across various architectures, making it suitable for efficient and reliable video decoding. The contents herein cover the functionalities of dav1d, including its installation, deployment via Ansible, and source code compilation.

dav1d supports the following features:

- support for all features of the AV1 bitstream
- support for all bitdepth, 8, 10 and 12bits
- support for all chroma subsamplings 4:2:0, 4:2:2, 4:4:4 and grayscale
- full acceleration for AVX-2 chips
- full acceleration for SSE3+ chips
- full acceleration for ARMv8 chips
- partial acceleration for ARMv7 chips

## 2 Links

- Homepage: <https://www.videolan.org/projects/dav1d.html>
- Source code: <https://code.videolan.org/videolan/dav1d>

## 3 Installation

To install dav1d on systems (tested under Debian 12 Bookworm) utilizing aptitude, the following commands are recommended:

### 1. Update the Package List

To ensure all dependencies are current, initiate an update with:

```
aptitude update
```

### 2. Install dav1d

Proceed with the installation of dav1d by executing:

```
aptitude install dav1d
```

### 3. Verify the Installation

Confirm the installation of dav1d by checking its version:

```
dav1d --version  
1.0.0
```

These steps facilitate the proper installation of dav1d, preparing the system for subsequent configuration and deployment tasks.

### 3.1 Ansible

```
- hosts: role_client
gather_facts: no
become: yes
vars:
  ns: video_nox
  packages:
    - dav1d
    - libdav1d6
tasks:
  - name: "{{ ns }}: Install and update packages"
    package:
      name: "{{ packages }}"
      state: latest
```

## 4 Configuration

dav1d is primarily focused on decoding AV1 video streams efficiently, and it does not involve extensive configuration at the user level since it's designed to be a fast and minimalistic decoder. The primary interaction users might have in terms of “configuration” typically involves choosing how to integrate and utilize dav1d in various applications or projects, rather than configuring the tool itself.

However, for the sake of thoroughness and to address situations where system libraries or dependencies like libdav1d6 (the dav1d library package) may involve configuration, the following section addresses this.

### 4.1 Configuration of dav1d and libdav1d6

dav1d is optimized for performance and simplicity and does not require traditional configuration files or settings adjustments post-installation. However, system administrators may need to ensure that it is correctly integrated with other video processing tools and applications.

#### 1. Library Integration

- Ensure that `libdav1d6` is recognized by software requiring AV1 decoding by verifying the linker's cache, this should not be needed under Debian.

```
ldconfig -p | grep libdav1d
libdav1d.so.6 (libc6,x86-64) =>
  ↪ /lib/x86_64-linux-gnu/libdav1d.so.6
```

- If the library is not listed, update the cache:

```
sudo ldconfig
```

## 2. Performance Tuning

- While dav1d itself does not offer configurable settings, performance can be influenced by the environment it operates within. Consider tuning system parameters such as CPU affinity and real-time priorities, especially in a multi-threaded environment, especially in multi CPU environments:

```
taskset -c [CPU cores] [mask] [command to run dav1d]
```

## 3. Integration with Media Players

- Ensure media players are configured to use dav1d for AV1 decoding. This usually involves setting dav1d as the preferred or default decoder for AV1 streams within the player's settings.

# 5 Usage

The dav1d command line tool includes the `--cpumask` option, which is designed to optimize decoding performance by restricting the usage of certain CPU instruction sets. This can be particularly useful in systems where specific instruction sets may lead to better decoding efficiency or in scenarios where it's necessary to maintain compatibility with older hardware that does not support newer instructions.

### Steps to Configure and Use `--cpumask` :

#### 1. Identify Supported Instruction Sets:

Before configuring the mask, determine which instruction sets are supported by the CPU. This can be done using the `lscpu` command or by examining `/proc/cpuinfo` :

```
grep flags /proc/cpuinfo | uniq
```

Example output:

```
1  flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
   pge mca \
2  cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx mmxext
   fxsr_opt \
3  pdpe1gb rdtscp lm constant_tsc rep_good nopl nonstop_tsc cpuid
   extd_apicid \
4  aperfmperf pni pclmulqdq monitor ssse3 fma cx16 sse4_1 sse4_2 popcnt
   aes \
5  xsave avx f16c lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a \
6  misalignsse 3dnowprefetch osvw ibs xop skinit wdt fma4 tce
   nodeid_msr tbm \
7  topoext perfctr_core perfctr_nb cpb hw_pstate ssbd ibpb vmcall bmi1
   arat \
8  npt lbrv svm_lock nrip_save tsc_scale vmcb_clean flushbyasid
   decodeassists \
9  pausefilter pfthreshold
10
```

## 2. Select the Appropriate Instruction Set:

Choose an instruction set that is supported by your CPU and offers the best performance for AV1 decoding. Common options include:

- `0` : No specific instruction set; use the default configuration
- `sse2` : Utilize the SSE2 instruction set
- `ssse3` : Utilize the SSSE3 instruction set
- `sse41` : Utilize the SSE4.1 instruction set
- `avx2` : Utilize the AVX2 instruction set for enhanced performance on newer CPUs
- `avx512` : Utilize the AVX512 instruction set for the highest performance on the latest CPUs.

## 3. Run dav1d with the Selected CPU Mask:

Once the optimal instruction set is identified, run dav1d with the `--cpumask` option. For example, to decode using the AVX2 instruction set, execute:

```
dav1d --cpumask=avx2 -i input.ivf --muxer=md5 -o output.md5
↳ # 1
dav1d --cpumask=avx2 -i input.ivf --muxer=yuv2mpeg -o
↳ output.yuv2mpeg # 2
dav1d --cpumask=avx2 -i input.ivf --muxer=yuv -o output.yuv
↳ # 3
```

1. MD5 checksum
2. Video playable

### 3. Raw video data

File	Size	Playable with vlc
source.mp4	16M	yes
input.ivf	14M	yes
output.md5	4.0K	no
output.yuv4mpeg2	268MB	yes
output.yuv	2.4G	no

Remark: Creating the `input.ivf` from an `input.mp4` can be done like so and can take a while:

```
ffmpeg -i input.mp4 -c:v libaom-av1 -crf 30 -b:v 0 input.ivf
```

(The `input.ivf` is actually the output of the `ffmpeg` command)

YUV files represent a category of file formats used primarily for storing raw video data. The YUV model separates the brightness information (luminance, denoted by Y) from the color information (chrominance, denoted by U and V) in a video.

Other interesting options (of 1.4.2-8-gda2cc78) are:

- `--limit N` stop decoding after N frames
- `--skip N` skip decoding of the first N frames
- `--threads N` number of threads (default: 0)
- `--verify MD5` verify decoded md5, implies `--muxer md5`, no output
- ...

## 6 Compiling Dav1d from Source

Compiling dav1d from source allows for customization and optimization specific to the system architecture in use. This section outlines the step-by-step process to compile the dav1d decoder. This should work Debian >= 10 Buster.

### 6.1 Prerequisites

Before starting the compilation process, ensure that the following dependencies are installed on the system:

- `meson` >= 0.49 - a build system required to configure the build environment.

- <https://mesonbuild.com/>
- <https://github.com/mesonbuild/meson>
- `ninja` - a small build system with a focus on speed.
  - <https://ninja-build.org/>
- `gcc` or `clang` - compilers that support C11.
- `git` - version control system to clone the repository.
- `nasm` (2.14 or higher) (requires Autoconf version 2.71 or higher, Debian 9 Stretch has 2.69-10) - General-purpose x86 assembler
- `autoconf` - General-purpose x86 assembler Install these dependencies using the package manager that corresponds with the system's distribution.

### 6.1.1 Installation

1. **Update Package Lists:** Before installing the packages, it's a good practice to update the package lists to ensure you are downloading the latest versions available. Run the following command:

```
aptitude update
```

2. **Install Required Packages:** Install the dependencies required to compile dav1d by executing the following command:

```
aptitude install meson ninja-build gcc git nasm autoconf
```

### 6.1.2 Verification

After the installation is complete, you can verify that the tools are installed correctly by checking their versions. Run the following commands:

- For Meson:

```
meson --version
1.0.1
ninja --version
1.11.1
gcc --version
gcc (Debian 12.2.0-14) 12.2.0
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There
↪ is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
↪ PURPOSE.
git --version
```

```
git version 2.39.2
```

## 6.2 Cloning the Repository

First, clone the dav1d repository from its official GitLab page:

```
git clone https://code.videolan.org/videolan/dav1d.git
cd dav1d
```

## 6.3 Configuring the Build

Use Meson to configure the build environment. A build directory should be created to keep the build separate from the source:

```
# The old form
meson build --buildtype release
# The new form
meson setup build --buildtype release
```

This command sets up the build directory with the configuration necessary for a release build, optimizing performance. Additional flags can be added to customize the build further.

## 6.4 Building the Project

After configuring the build, use Ninja to compile dav1d. Just run your build command (e.g. `ninja`) and Meson will regenerate as necessary. If `ninja` fails, run “`ninja reconfigure`” or “`meson setup --reconfigure`” to force Meson to regenerate.

If build failures persist, run “`meson setup --wipe`” to rebuild from scratch using the same options as passed when configuring the build. To change option values, run “`meson configure`” instead.

```
ninja -C build
```

This command compiles the code using the settings specified in the `meson` configuration step. After compilation dav1d can be found under `build/tools`.

## 6.5 Installing

Once the build is complete, install the binary to the system:



```
ninja -C build install
```

## 6.6 Verification

To ensure the binary was compiled and installed correctly, run:

```
dav1d --version;build/tools/dav1d --version
1.0.0 # system version
1.4.2-8-gda2cc78 # compiled version
```

## 7 History

Version	Date	Notes
0.1.1	2024-05-30	Compilation from source (Quick Guide release)
0.1.0	2023-03-08	Initial release

## 8 Disclaimer of Warranty

THERE IS NO WARRANTY FOR THIS INFORMATION, DOCUMENTS AND PROGRAMS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE INFORMATION, DOCUMENT OR THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE INFORMATION, DOCUMENTS AND PROGRAMS IS WITH YOU. SHOULD THE INFORMATION, DOCUMENTS OR PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

## 9 Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE INFORMATION, DOCUMENTS OR PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION, DOCUMENTS OR PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE INFORMATION, DOCUMENTS OR PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.