

## Build Steps for LAME MP3 Encoder for Windows-ARM64

Build toolchain used: MSYS2 CLANGARM64

### 1.1.1. Tool required:

- Msys2 - MSYS2 is a collection of tools and libraries providing you with an easy-to-use environment for building, installing and running native Windows software.
- Download: [Download Msys2](#)
- Installation procedure: [MSYS2 Installation](#)

### 1.1.2. Steps to build Lame in ARM64 Native:

Download lame source from: [LAME \(Lame Aint an MP3 Encoder\) - Browse /lame/3.100 at SourceForge.net](#)

- Open MSYS2 CLANGARM64 application from start menu.
- Run below commands for setup:

```
1. pacman -Suy
2. pacman -S mingw-w64-clang-aarch64-clang
3. pacman -S make
4. pacman -S mingw-w64-clang-aarch64-gtk3
5. pacman -S mingw-w64-clang-aarch64-icu
6. pacman -S mingw-w64-clang-aarch64-harfbuzz
```

- Go to LAME source directory and build:

```
1. cd <lame_src_dir>
```

- Download the patch files attached in the attachment section:

```
1. parse_changes.patch
2. sym_changes.patch
```

- Apply patches from the root directory of lame and build lame using Make:

```
1. patch frontend/parse.c < parse_changes.patch
2. patch include/libmp3lame.sym < sym_changes.patch
```

- Download the patch for Lame MP3 neon optimization from [here](#) and save it as neon\_changes.patch.
- Now, apply the patch from the root directory of lame using the following command:

```
1. patch -p0 < neon_changes.patch
```

- Now, open the root directory of the project in Visual Studio Code, search for “#if defined(\_\_aarch64\_\_) || defined(\_\_arm\_\_)” and replace it with “#if defined(\_\_aarch64\_\_) || defined(\_M\_ARM64)”

- Build the project and verify executable type generated with the below command in Msys2 Clangarm64 terminal:

```
./configure  
make  
file frontend/lame.exe
```