

# ARM and AArch64 Target Triples

## Introduction

The GCC compiler uses '*triples*' when referring to the host system that the compiler runs on or the target system that a compiler will generate binaries to run on.

The term '*triple*' refers to the method used in the GNU/Linux operating system to designate a specific host or target. It is separated into the following components:

```
<arch>-<vendor>-<operatingsystem>
```

The *<operatingsystem>* may be further divided into an optional *operating system* and *ABI* portions. For example the Linux OS and the GNU hardfloat ABI might have the following *<operatingsystem>*, "-linux-gnueabi". Whereas a bare-metal target will not have an OS but only the embedded ABI, "-eabi".

<https://gcc.gnu.org/install/specific.html>

For default configuration of each '*triples*', see [ARM and AArch64 Target Triples default options](#).

## Little-Endian Triples

Little-endian is the default bit ordering on ARM.

### aarch64-linux-gnu

Toolchains for little-endian, 64-bit **ARMv8** for GNU/Linux systems

```
gcc-linaro-*x86_64_aarch64-linux-gnu.tar.xz  
    Linux 64-bit binaries for the Aarch64 Linux cross-toolchain  
gcc-linaro-*i686_aarch64-linux-gnu.tar.xz  
    Linux 32-bit binaries for the Aarch64 Linux cross-toolchain  
gcc-linaro-*i686-mingw32_aarch64-linux-gnu.tar.xz  
    Windows 32-bit binaries for the Aarch64 Linux cross-toolchain
```

Default command line options (gcc -v -c): '-march=armv8-a' '-mlittle-endian' '-mabi=lp64'

### arm-linux-gnueabi

Toolchains for little-endian, hard-float, 32-bit **ARMv7** (and earlier) for GNU/Linux systems

```
gcc-linaro-*x86_64_arm-linux-gnueabi.tar.xz  
    Linux 64-bit binaries for the ARMv7 Linux hard float cross-toolchain  
gcc-linaro-*i686_arm-linux-gnueabi.tar.xz  
    Linux 32-bit binaries for the ARMv7 Linux hard float cross-toolchain  
gcc-linaro-*i686-mingw32_arm-linux-gnueabi.tar.xz  
    Windows 32-bit binaries for the ARMv7 Linux hard float cross-toolchain
```

Default command line options (gcc -v -c): '-march=armv7-a' '-mtune=cortex-a9' '-mfloat-abi=hard' '-mfpu=vfpv3-d16' '-mthumb' '-mtdialect-gnu'

### armv8l-linux-gnueabi

Toolchains for little-endian, 32-bit **ARMv8** for GNU/Linux systems

```
gcc-linaro-*x86_64_armv8l-linux-gnueabi.tar.xz  
    Linux 64-bit binaries for the Aarch64 Linux cross-toolchain  
gcc-linaro-*i686_armv8l-linux-gnueabi.tar.xz  
    Linux 32-bit binaries for the Aarch64 Linux cross-toolchain  
gcc-linaro-*i686-mingw32_armv8l-linux-gnueabi.tar.xz  
    Windows 32-bit binaries for the Aarch64 Linux cross-toolchain
```

Default command line options (gcc -v -c): '-march=armv8-a' '-mfloat-abi=hard' '-mfpu=neon-fp-armv8' '-mthumb' '-mtdialect-gnu'

### arm-linux-gnueabi

Toolchains for little-endian, soft-float, 32-bit **ARMv7** (and earlier) for GNU/Linux systems

```
gcc-linaro-*x86_64_arm-linux-gnueabi.tar.xz
    Linux 64-bit binaries for the ARMv7 Linux soft float cross-toolchain
gcc-linaro-*i686_arm-linux-gnueabi.tar.xz
    Linux 32-bit binaries for the ARMv7 Linux soft float cross-toolchain
gcc-linaro-*i686-mingw32_arm-linux-gnueabi.tar.xz
    Windows 32-bit binaries for the ARMv7 Linux soft float cross-toolchain
```

Default command line options (gcc -v -c): '-march=armv7-a' '-mtune=cortex-a9' '-mfloat-abi=soft' '-mthumb' '-mtls-dialect=gnu'

## aarch64-elf

Toolchains for little-endian, 64-bit **ARMv8** for bare-metal systems

```
gcc-linaro-*x86_64_aarch64-elf.tar.xz
    Linux 64-bit binaries for the Aarch64 bare-metal cross-toolchain
gcc-linaro-*i686_aarch64-elf.tar.xz
    Linux 32-bit binaries for the Aarch64 bare-metal cross-toolchain
gcc-linaro-*i686-mingw32_aarch64-elf.tar.xz
    Windows 32-bit binaries for the Aarch64 bare-metal cross-toolchain
```

Default command line options (gcc -v -c): '-march=armv8-a' '-mlittle-endian' '-mabi=lp64'

## arm-eabi

Toolchains for little-endian, soft-float, 32-bit **ARMv7** (and earlier) for bare-metal systems

```
gcc-linaro-*x86_64_arm-eabi.tar.xz
    Linux 64-bit binaries for the ARMv7 bare-metal cross-toolchain
gcc-linaro-*i686_arm-eabi.tar.xz
    Linux 32-bit binaries for the ARMv7 bare-metal cross-toolchain
gcc-linaro-*i686-mingw32_arm-eabi.tar.xz
    Windows 32-bit binaries for the ARMv7 bare-metal cross-toolchain
```

Default command line options (gcc -v -c): None, but 'gcc -S -v' produces '.cpu arm7tdmi' and '.fpu softvfp' in the .s file

## Big-Endian Triples

### aarch64\_be-linux-gnu

Toolchains for big-endian, 64-bit **ARMv8** for GNU/Linux systems

```
gcc-linaro-*x86_64_aarch64_be-linux-gnu.tar.xz
    Linux 64-bit binaries for the Aarch64 Linux Big Endian cross-toolchain
gcc-linaro-*i686_aarch64_be-linux-gnu.tar.xz
    Linux 32-bit binaries for the Aarch64 Linux Big Endian cross-toolchain
gcc-linaro-*i686-mingw32_aarch64_be-linux-gnu.tar.xz
    Windows 32-bit binaries for the Aarch64 Linux Big Endian cross-toolchain
```

Default command line options (gcc -v -c): '-march=armv8-a' '-mbig-endian' '-mabi=lp64'

### armeb-linux-gnueabi

Toolchains for big-endian, hard-float, 32-bit **ARMv7** (and earlier) for GNU/Linux systems

```
gcc-linaro-*x86_64_armeb-linux-gnueabi.tar.xz
    Linux 64-bit binaries for the ARMv7 Linux Big Endian hard float cross-toolchain
gcc-linaro-*i686_armeb-linux-gnueabi.tar.xz
    Linux 32-bit binaries for the ARMv7 Linux Big Endian hard float cross-toolchain
gcc-linaro-*i686-mingw32_armeb-linux-gnueabi.tar.xz
    Windows 32-bit binaries for the ARMv7 Linux Big Endian hard float cross-toolchain
```

Default command line options (gcc -v -c): '-march=armv7-a' '-mtune=cortex-a9' '-mfloat-abi=hard' '-mfpu=vfpv3-d16' '-mthumb' '-mtls-dialect=gnu'

### armeb-linux-gnueabi

Toolchains for big-endian, soft-float, 32-bit **ARMv7** (and earlier) for GNU/Linux systems

gcc-linaro-\*x86\_64\_armeb-linux-gnueabi.tar.xz  
Linux 64-bit binaries for the ARMv7 Linux Big Endian soft float cross-toolchain  
gcc-linaro-\*i686\_armeb-linux-gnueabi.tar.xz  
Linux 32-bit binaries for the ARMv7 Linux Big Endian soft float cross-toolchain  
gcc-linaro-\*i686-mingw32\_armeb-linux-gnueabi.tar.xz  
Windows 32-bit binaries for the ARMv7 Linux Big Endian soft float cross-toolchain

Default command line options (gcc -v -c): '-march=armv7-a' '-mtune=cortex-a9' '-mfloat-abi=soft' '-mthumb' '-mtls-dialect=gnu'

## aarch64\_be-elf

Toolchains for big-endian, 64-bit **ARMv8** for bare-metal systems

gcc-linaro-\*x86\_64\_aarch64\_be-elf.tar.xz  
Linux 64-bit binaries for the Aarch64 bare-metal Big Endian cross-toolchain  
gcc-linaro-\*i686\_aarch64\_be-elf.tar.xz  
Linux 32-bit binaries for the Aarch64 bare-metal Big Endian cross-toolchain  
gcc-linaro-\*i686-mingw32\_aarch64\_be-elf.tar.xz  
Windows 32-bit binaries for the Aarch64 bare-metal Big Endian cross-toolchain

Default command line options (gcc -v -c): '-march=armv8-a' '-mbig-endian' '-mabi=lp64'

## armeb-eabi

Toolchains for big-endian, soft-float, 32-bit **ARMv7** (and earlier) for bare-metal systems

gcc-linaro-\*x86\_64\_armeb-eabi.tar.xz  
Linux 64-bit binaries for the ARMv7 bare-metal Big Endian cross-toolchain  
gcc-linaro-\*i686\_armeb-eabi.tar.xz  
Linux 32-bit binaries for the ARMv7 bare-metal Big Endian cross-toolchain  
gcc-linaro-\*i686-mingw32\_armeb-eabi.tar.xz  
Windows 32-bit binaries for the ARMv7 bare-metal Big Endian cross-toolchain

Default command line options (gcc -v -c): None, but 'gcc -S -v' produces '.cpu arm7tdmi' and '.fpu softvfp' in the .s file