



Apertis Toolchain

1 Contents

2 Usage of the toolchain 2

3 The Apertis Toolchain provides a build environment to build Apertis for a for-
4 eign architecture from an operating system that is not Apertis.

5 The toolchain is intended for cross-compilation, targeting architectures that
6 don't match the CPU architecture of the build host (for instance, building
7 ARM64 binaries from a Intel-based host). Currently Apertis provides toolchain
8 for two architectures :

- 9 • 32 bits ARM (armhf)
- 10 • 64 bits ARM (AArch64)

11 It is meant to be used by teams that want to keep their development tools such
12 as their preferred editor.

13 The toolchain only provides a compiler and minimal libraries to build code.
14 The main supported language is C, but C++ is also supported. The toolchain
15 provides a debugger with multiarch support for debugging foreign binaries. To
16 build binaries using other libraries, either those libraries have to be built with
17 the toolchain, or the developer can use a sysroot containing the libraries.

18 Sysroots are file system trees specifically meant for cross-compilation and remote
19 debugging targeting a specific release image.

20 See [Sysroots and devroots](#)¹ for more information.

21 Usage of the toolchain

22 The toolchain for AArch64 and armhf can be downloaded on the [Apertis release](#)
23 [site](#)².

24 We recommend following the [guidance on product development on Apertis](#)³
25 when picking which release to use.

26 To extract the toolchain for AArch64 use the following commands:

```
27 $ wget https://images.apertis.org/release/v2020/v2020.2/toolchain/apertis-  
28 aarch64-linux-gnu-toolchain.tar.xz  
29 $ xz -d apertis-aarch64-linux-gnu-toolchain.tar.xz  
30 $ tar xf apertis-aarch64-linux-gnu-toolchain.tar  
31 $ cd apertis-aarch64-linux-gnu-toolchain/
```

32 Download and extract a sysroot. Ignore errors related to `mknod`. Be careful
33 about the extraction of the sysroot. There is no directory prepended to the

¹<https://martyn.pages.apertis.org/apertis-website/architecture/sysroots-and-devroots/>

²<https://images.apertis.org/release>

³<https://martyn.pages.apertis.org/apertis-website/policies/release-flow/#guidelines-for-product-development-on-top-of-apertis-and-its-direct-downstreams>

34 extraction path of the sysroot archive. It can become problematic if extracted
35 in the HOME directory. Make sure to create a work directory and change to it
36 before extraction.

```
37 $ wget https://images.apertis.org/release/v2020/v2020.2/arm64/sysroot/sysroot-  
38 apertis-v2020-arm64-v2020.2.tar.gz  
39 mkdir ~/sysroot/  
40 tar xf sysroot-apertis-v2020-arm64-v2020.2.tar.gz -C ~/sysroot/ --  
41 exclude=./dev
```

42 Retrieve the project to build:

```
43 git clone git@gitlab.apertis.org:tests/bluez-phone-tester.git
```

44 From there, the steps are project dependent. Not all projects support cross
45 compilation setup. It can usually be encompassed using environment variables.

```
46 $ ./autogen.sh  
47 $ CC=~/.apertis-aarch64-linux-gnu-toolchain/usr/bin/aarch64-linux-gnu-gcc-  
48 8 PKG_CONFIG_SYSROOT_DIR=~/.sysroot PKG_CONFIG_PATH=~/.sysroot/usr/lib/aarch64-  
49 linux-gnu/pkgconfig/ ./configure --host aarch64-linux-gnu  
50 $ make CFLAGS="-I ~/.sysroot/usr/include" LDFLAGS="-L ~/.sysroot/usr/lib/aarch64-  
51 linux-gnu -L ~/.apertis-aarch64-linux-gnu-toolchain/usr/lib/../../ -L ~/.sysroot/usr/aarch64-  
52 linux-gnu/lib/ --sysroot=~/.sysroot"  
53 $ file bluez-phone-tester  
54 bluez-phone-tester: ELF 64-bit LSB executable, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /  
55 linux-aarch64.so.1, for GNU/Linux 3.7.0, BuildID[sha1]=ca3fb7801ab26632208a6def82ac1dcf5cb40d10, not stripped
```

56 This project did not require specific dependencies, everything needed was in the
57 sysroot. The sysroot are meant to be self sufficient for building the Apertis
58 projects. If a new dependency is needed in the sysroot, the recipe should be
59 modified. It is explicitly discouraged to install dependencies using different
60 means because there should not be absolute symbolic links in the sysroot since
61 they would not be pointing to the sysroot root folder. This has the potential to
62 damage the system.