Introduction

Achieve deterministic builds

Define a build environment

Distribute the environment

Recording the build environment

It is been customary in user facing software to provide a way for developers investigating bugs to learn how the software has been built. The “about dialog” or output of `--version` typically contains information about the build environment.

In the context of reproducible builds, we either actively make aspects of the build environment irrelevant to the build output, or ensure they are available to rebuild the software exactly as distributed.

All relevant information about the build environment should either be defined as part of the development process or recorded during the build process.

File Format

Everything that is recorded is stored best as a separate build product that can be easily ignored or distributed separately. This will help identify which variation is irrelevant to the software itself.

This product is called the ‘buildinfo’, but its exact format and the way it is distributed differs across ecosystems.

Debian

Debian shares its buildinfo files as plain text files following the control file format, usually clearsigned with OpenPGP. A detailed description of the expected fields and values, as well as conventions around naming, can be found under

ReproducibleBuilds/BuildinfoFiles on the Debian wiki. Examples can be found on buildinfo.debian.net.

Arch Linux

The Arch Linux makepkg build tool produces a `.BUILDINFO` file consisting of `<key> = <value>` pairs.

Unlike on Debian, this file is not independently signed and distributed, but included into the package (and thus signed as part of the package signature). An example can be found by downloading any Arch package built with a recent version of makepkg, such as archlinux-keyring.

Tails

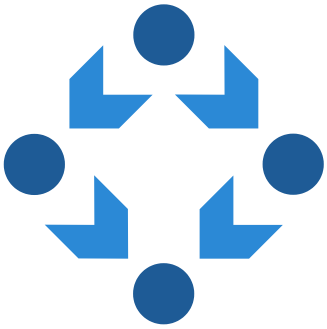
Tails does not record a buildinfo file per se, but instead the vagrant directory of the main git repo contains all information necessary to reproducibly rebuild that revision of Tails.

JVM

In the JVM ecosystem it is common to distribute libraries as binary (bytecode) jars uploaded to a repository like Maven Central or Google's Android Repository.

It is recommended that the buildinfo describing the build environment used for that official build is published alongside each artifact. Third party attestations can be shared in a separate sig-repo. For a detailed overview of the conventions so far see the separate JVM page.

BACK NEXT



Reproducible Builds

"Reproducible builds" aim to provide a verifiable path from software source code to its compiled binary form.

@ReproBuilds

Projects working on reproducible builds:
Arch Linux, Baserock, Bitcoin, coreboot, Debian, ElectroBSD, F-Droid, FreeBSD, Fedora, GNU Guix, Monero, NetBSD, NixOS, OpenEmbedded, openSUSE, OpenWrt, Qubes OS, Symfony, Tails, Tor Browser, Webconverger, Yocto Project.
Please tell us about yours!

Patches highly welcome through our Git repository (more info) or via our mailing list.