

bullet

The Bullet Linux distribution's repository

Components

- Bullet Repository ~ build scripts for packages in the Stable and Lab channel
- Bullet Templates ~ templates for Bullet packages. Also contains `pkmake.sh`, an utility for creating new packages.
- ABS (Automated Build System) ~ program for building packages and images for Bullet Linux
- `bpkg` ~ Bullet Linux's package manager

Possible future Projects (all those projects are independent from bullet, and they will be)

- AIS (Automated Install System) ~ Easy but flexible system for automatically installing Bullet Linux on different machines.
- `coffee-init` ~ Fuck `systemd`.
- `nic` (Nic is not CUPS) ~ Fuck CUPS.
- `reblued` ~ (Reliable bluetooth daemon). Fuck BlueZ.
- `neoland` ~ Wayland compositor. Fully customizable and re-programmable in Lisp.

Branches

There are two branches: Stable and Lab.

Lab is used mostly by maintainers; packages get updated as soon as they are released from the authors, and they can be really unstable; not only that, but they can also contain ABS errors, which means that they are packages that either fail their required tests or can be cause of dependency collision, in fact all those issues must be resolved before those packages can be sent to Stable. The normal wait time is 1-2 weeks for minor releases, 2-8 weeks for major releases. It's a good idea to use only in containers/VMs, and test hardware-related packages (like the kernel) using a real machine running Bullet Linux in the Stable branch and locking those packages in the Lab branch. Stable is useable everybody; it's a rolling release but it's REALLY stable, thanks to the package requirements, the small size of the repository and the period of time that each packages has to pass in the Lab branch. If you don't know which one to use, please use the Stable.

Lab+

If you want to use Lab with less unstable packages, you can try to enter PSB-free mode with `bpkg psb-free on`. (You can also disable it with `bpkg psb-free off`)

PSB stands for Possible System Breaker, and it's a flag that is enabled on packages that either don't pass all the implemented test or can be cause of dependency collision. The PSB-free mode will stop the package manager from updating your packages to any PSB-flagged version.

NOTE: The Archive for the Lab branch will always keep the last non-PSB version of any package, alongside the latest one.

Concepts

- Small collection of packages = packages are more curated and tested
- Packages get updated in Lab as soon as possible (rolling release design).

Stability

- Bulk updates (ex. if lib-x and program-y get a new update, and program-y requires the new version of lib-x, we will always make the new versions available at the same time, so that it isn't possible to partially update the system,).
- ABS checks globally the conditions of all dependencies, so that the maintainers will get notified of condition that can (and will) lead to "dependency hell", so that it can be fixed.
- When an library gets updated, all packages that depends on it will be recompiled against the new version of the library. (This brings more stability)
- ATO (Automated Testing Oriented) -> Packages can implement a `test` command, which can be useful with unit testing. This will than be used in order to test if packages are well working before going to the stable or in some cases before going to Lab.

bpkg

- A simple, flexible, compatible, reproducible and recoverable package manager:
 - Simple: `install`, `remove`, `clean`, `update`. Is this easy enough?
 - Flexible:
 - Compatible: you can download binaries from Bullet Archives, compile official packages from the Bullet Repository, compile community-provided packages from the Community Bullet Repository, you can directly install `.deb` files or install them, with their dependencies (if they aren't found in the Bullet Repository) from any deb-compatible remote mirror (ex. Debian, Ubuntu, etc...).
 - Reproducible: Bpkg allows you to export the entire or part of the package tree (with the version of the packages), import it in another system in order to have an exact copy, embed it in packages in order to have the advantages of Docker (without being a container), and even building ISO and OCI images.

- Recoverable: Bpkg takes inspiration from git; in fact,

HowTo

How to install the local Toolchain

NOTE: `bpkg` and `ABS` are required to be locally installed in order to build Bullet Linux. They can be installed either system-wide (in `/usr/bin`) or user-wide (in `$HOME`).

- 1) Install the Rust compiler.
- 2) Clone/pull this repository.
- 3) `./install.sh` (requires `sudo` if you want to install for all users)

How to build the rootfs

For Stable releases: - Build Rootfs: `abs rootfs repos/stable repos/stable.list rootfs.tar`
For Lab releases: - Build Rootfs: `abs rootfs repos/lab repos/lab.list rootfs.tar`

How to build images

After that, you can build: - ISO image: `abs iso rootfs.tar bullet.iso` -
OCI image: `abs oci rootfs.tar bullet.oci`