Pacman (Package Manager)

Underneath the surface, most things in updates & downloads are powered by pacman, which if you've used Arch Linux before are probably familiar with. Using pacman, you can easily install/upgrade/remove your own packages, or packaged provided and hosted by the Batocera community. The exception to this is Batocera's updater.

It is important to note that although this is pacman, and Arch Linux uses pacman to install *its* packages, it does not mean that you can install packages designed for Arch Linux inside of Batocera. It's the same tool, but the packages themselves must be made to be compatible with Batocera, not just Arch Linux/other distribution that uses pacman.

Obviously, packages hosted by Batocera **will never include copyrighted material** – only freeware or shareware ROMs, homebrews, free games, themes, free musical themes and so on. It's a way to easily customize your Batocera to fit better with your needs.

The advantages of using a package manager are:

- one package contains all the files needed: ROM, scraped metadata, update in the gamelist.xml…
- you can install packages directly from the Internet
- you can update them when a new release of the package is available
- you can easily remove packages that you don't want any longer
- dependencies are automatically resolved

Pacman was chosen because it is lightweight, very portable, and with Batocera's version, you can package your own pacman packages on your Batocera box **without a full dev environment**. Batocera provides its specific tools (slimmed down from ArchLinux's full pacman distribution).

From a technical perspective, a pacman package is simply a `tar.zstd` or `.tar.xz` archive – exact same format as Batocera's `boot.tar.xz`. Newer versions of the packages use zstd rather than xz as the compression algorithm, as it's much faster to decompress, especially for larger packages.

Pacman via SSH

Batocera provides a full UI in EmulationStation to manage your pacman packages from **UPDATES & DOWNLOADS → CONTENT DOWNLOADER**. As an alternative to this, you can utilize the `batocera-store` command line tool. Here are its list of commands:

```
/usr/bin/batocera-store install <package>
/usr/bin/batocera-store remove  <package>
/usr/bin/batocera-store list
/usr/bin/batocera-store list-repositories
/usr/bin/batocera-store clean
/usr/bin/batocera-store clean-all
/usr/bin/batocera-store refresh
```

```
/usr/bin/batocera-store update
```

The first two commands are self-explanatory; `list` lists all the packages you can install, and `list-repositories` lists all the repositories from where packages can be installed.

Whenever a package is downloaded and installed, pacman keeps a local cache of the package into `/userdata/system/pacman/pkg/`. The command `clean` will clean up this repository and keep only the last version installed, while `clean-all` will remove all files from the cache.

Finally `refresh` will refresh the packages available from all the repositories, and `update` will update all your installed packages to their latest version.

How can I use pacman as a user?

Users usually need only `batocera-store` commands. But here are some details on how it works under the hood.

The first thing to do is to synchronize your local pacman DB with Batocera's repository (kind of an “app store” for packages).

```
 # pacman -Sy
 :: Synchronizing package databases...
 batocera                768.0   B  0.00   B/s 00:00 [----------------------
------------------------] 100%
```

Then you can search for all packages available

```
 # pacman -Ss
 batocera/bezels-default-glazed 1.0.0-1
     Batocera bezels with a CRT effect
 batocera/nes-alter-ego 1.0.0-1
     NES freeware puzzle game - manual provided as a pdf
 batocera/ports-quake-shareware 1.0.0-1
     Quake game files (shareware version)
```

You can search from string, like `pacman -Ss bezels`.

In order to install/upgrade a package you can type:

```
 # pacman -S nes-alter-ego
 resolving dependencies...
 looking for conflicting packages...
 Packages (1) nes-alter-ego-1.0.0-1
 Total Download Size:   0.40 MiB
 Total Installed Size:  0.49 MiB
 :: Proceed with installation? [Y/n]
 (1/1) checking keys in keyring                     [----------------------
------------------------] 100%
 (1/1) checking package integrity                   [----------------------
```

```
------------------------] 100%
 (1/1) loading package files                          [----------------------
------------------------] 100%
 (1/1) checking for file conflicts                    [----------------------
------------------------] 100%
 (1/1) checking available disk space                  [----------------------
------------------------] 100%
 (1/1) reinstalling nes-alter-ego                     [----------------------
------------------------] 100%
 :: Running post-transaction hooks...
 (1/1) batocera-install.hook
 Entry for Alter_Ego.nes added in /userdata/roms/nes/gamelist.xml
```

You game will be added to EmulationStation, and you can play this newly installed game!

To remove a package:

```
 # pacman -R nes-alter-ego
 checking dependencies...
 Packages (1) nes-alter-ego-1.0.0-1
 Total Removed Size:  0.47 MiB
 :: Do you want to remove these packages? [Y/n]
 :: Processing package changes...
 (1/1) removing nes-alter-ego                          [----------------------
------------------------] 100%
```

Install, remove and manage packages

The easiest is to install a package from a repository. For example, to install `nes-alter-ego` from the Batocera repository, just enter:

```
pacman -S nes-alter-ego
```

It is not necessary to setup a repository to test your own packages (see below about setting up your own repositories). You can simply install or upgrade a local package in .tar.xz format by typing:

```
 pacman -U mypackage-1.0.0-1.tar.xz
```

To remove a package, whether it's been installed from a repository or from an individual package, you need to use the `-R` switch, without providing the version of the package that is intalled. For example:

```
 pacman -R mypackage
```

A list of all currently installed packages can be seen with the following:

```
 pacman -Q
```

Also, you can see that every package that gets installed (whether from an individual package or from a repository) is copied locally to `/userdata/system/pacman/pkg/` on your Batocera system. This

local cache directory of packages can be cleaned up, i.e. remove all the package files, by entering:

```
pacman -Scc
```

Manage your own repository

You can use pacman to manage your own library of packages (if you want to add your own ROMs to the free ones distributed by the Batocera game store for example).

In order to do so, you can create a local repository by adding a new section in `/userdata/system/pacman/pacman.conf`:

```
[private_repo]
Server            = file:///userdata/local_repo/
```

Server can be a local directory like `/userdata/local_repo/` or you can host that on your own web server.

To add a package to your local repository, copy your new package `mypackage-1.0.0-1.tar.xz` to where your local repository stands and type:

```
# repo-add /userdata/local_repo/private_repo.db.tar.gz mypackage-1.0.0-1-
any.pkg.tar.xz
==> Extracting private_repo.db.tar.gz to a temporary location...
==> Extracting private_repo.files.tar.gz to a temporary location...
==> Adding package 'mypackage-1.0.0-1-any.pkg.tar.xz'
  -> Computing checksums...
  -> Creating 'desc' db entry...
  -> Creating 'files' db entry...
==> Creating updated database file
'/userdata/local_repo/private_repo.db.tar.gz'
```

Once the package is added to your repo, synchronize your pacman with the latest information by typing `pacman -Sy` and you can now install your package from your repository.

To remove a package from your repository, you can use `repo-remove /userdata/local_repo/private_repo.db.tar.gz mypackage` (it's the name of the package, not the file name with the version).

Make your own package

For that, refer to the "Create your own pacman package" developer page about it.

From:
https://wiki.batocera.org/ - **Batocera.linux - Wiki**

Permanent link:
https://wiki.batocera.org/pacman_package_manager

Last update: **2022/08/10 03:41**