



Dpkg

Dpkg



A quality version of this page, approved on *9 March 2020*, was based off this revision.

Contents

1 Article purpose	3
2 Introduction	4
3 Getting Started	5
4 References	6



1 Article purpose

This article aims to give information required to start with the Linux tool: dpkg^[1]



2 Introduction

`dpkg` is the basic tool to deal with this package manager. A package manager is helpful for installing, removing and updating package in the rootfs.

A package is an archive with `.deb` extension that contains a set of files but also a set of installing/removing scripts.

Any package provides two kind of artifacts:

- metadata: collection of scripts and configs needs to correctly install data of the package
- data: applications, libraries, and any files to copy to the rootfs



3 Getting Started

dpkg can be very helpful to know which file of the rootfs belongs to which package.

Keep in mind that the OpenSTLinux distribution has been generated with OpenEmbedded. There is a direct parity between 'recipes' and 'packages'.

Indeed an OpenEmbedded recipe describes how to get sources code, how to compile it, where to install built artifacts on filesystem and also which 'dpkg' packages to create and which files to put in.

That is why it can be very interesting to list the files that belong to a package or to find a package that belongs to a file.

Here a short list of useful commands with 'dpkg':

- How to list the currently installed packages

```
$> dpkg -l
```

- How to install a package:

```
$> dpkg -i <packageName>.deb
```

- How to remove/uninstalling a package

```
$> dpkg -r <packageName>
```

- How to list files of a package

```
$> dpkg -L <packageName>
```

- How to find a package that belongs to a file

```
$> dpkg -S </absolute/path/to/a/file/of/target>
```



4 References

- <https://linux.die.net/man/1/dpkg>

Linux® is a registered trademark of Linus Torvalds.