Add Packages to an Image

Using the IMAGE_INSTALL variable

Use IMAGE_INSTALL to add **packages** to an existing image

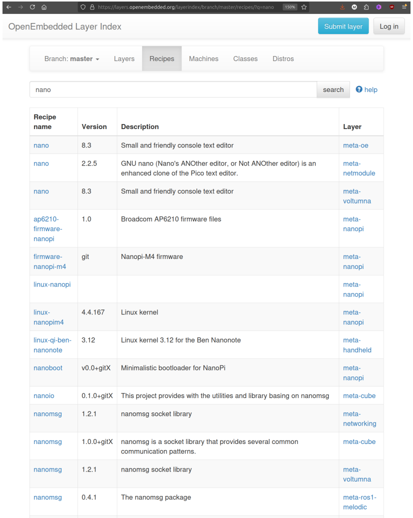
- Remember that packages are installed, not recipes
- You don't need all the packages produced by a recipe
- However, very often the main binary package name coincides with the recipe name
- Typically appended in conf/local.conf. Examples:

```
IMAGE_INSTALL:append = " os-release curl"
```

```
IMAGE_INSTALL:append = " libflac"  # Just libflac, not the flac executable
```

Find the recipe(s) you need

- Go to `https://layers.openembedded.org/`
- Click on the `Recipes` tab
- Make a search by recipe name
- You get the matching recipes and the layer they belong to.

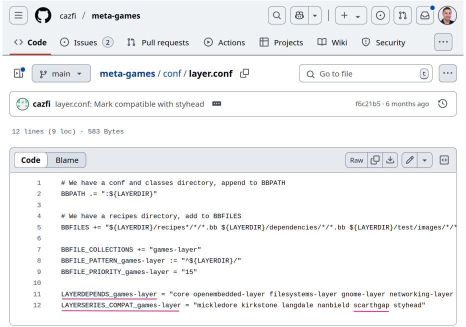


Recipe name	Version	Description	Layer
nano	8.3	Small and friendly console text editor	meta-oe
nano	2.2.5	GNU nano (Nano's ANOther editor, or Not ANOther editor) is an enhanced clone of the Pico text editor.	meta-netmodule
nano	8.3	Small and friendly console text editor	meta-voltumna
ap6210-firmware-nanopi	1.0	Broadcom AP6210 firmware files	meta-nanopi
firmware-nanopi-m4	git	Nanopi-M4 firmware	meta-nanopi
linux-nanopi			meta-nanopi
linux-nanopim4	4.4.167	Linux kernel	meta-nanopi
linux-qi-ben-nanonote	3.12	Linux kernel 3.12 for the Ben Nanonote	meta-handheld
nanoboot	v0.0+gitX	Minimalistic bootloader for NanoPi	meta-nanopi
nanoio	0.1.0+gitX	This project provides with the utilities and library basing on nanomsg	meta-cube
nanomsg	1.2.1	nanomsg socket library	meta-networking
nanomsg	1.0.0+gitX	nanomsg is a socket library that provides several common communication patterns.	meta-cube
nanomsg	1.2.1	nanomsg socket library	meta-voltumna
nanomsg	0.4.1	The nanomsg package	meta-ros1-melodic

Install additional layers

- Stay on `https://layers.openembedded.org/`, and visit the link for the layer your recipe belongs to.
- Follow the link in the web repo button.
- In the web repository, follow the link to the `conf/layer.conf` file.
- Check the `LAYERDEPENDS` and `LAYERSERIES_COMPAT` for additional layers, and for the compatibility with the Yocto/OE branch you're using.
- Clone the source code of your layer.
- Add your layer to `conf/bblayers.conf`:

```
$ bitbake-layers add-layer <path-to-layer>
```



cazfi layer.conf: Mark compatible with styhead f6c21b5 · 6 months ago

12 lines (9 loc) · 583 Bytes

```
# We have a conf and classes directory, append to BBPATH
BBPATH .= ":${LAYERDIR}"

# We have a recipes directory, add to BBFILES
BBFILES += "${LAYERDIR}/recipes*/*/*.bb ${LAYERDIR}/dependencies/*/*.bb ${LAYERDIR}/test/images/*/*

BBFILE_COLLECTIONS += "games-layer"
BBFILE_PATTERN_games-layer := "^${LAYERDIR}/"
BBFILE_PRIORITY_games-layer = "15"

LAYERDEPENDS_games-layer = "core openembedded-layer filesystems-layer gnome-layer networking-layer
LAYERSERIES_COMPAT_games-layer = "mickledore kirkstone langdale nanbield scarthgap styhead"
```

Find the package(s) you need

You first need to build the recipe you are interested in:

```
bitbake flac
```

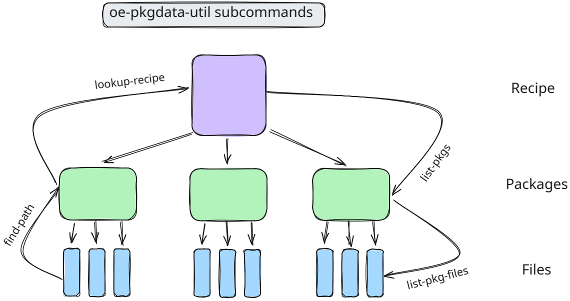
Then, you can query the packages that have been built:

```
$ oe-pkgdata-util list-pkgs -p flac
flac
flac-dbg
flac-dev
flac-doc
flac-src
libflac
libflac++
```

oe-pkgdata-util script

`oe-pkgdata-util`:
Very handy script to query package information
(without having to install a package manager on the target).

- `oe-pkgdata-util find-path <path>`
  Find the package providing a path in the image
- `oe-pkgdata-util lookup-recipe <package>`
  Find the recipe implementing the package
- `oe-pkgdata-util list-pkgs -p <recipe>`
  List packages built by a recipe
- `oe-pkgdata-util list-pkg-files <package>`
  List files belonging to a package



oe-pkgdata-util script — Example

```
$ oe-pkgdata-util list-pkgs -p flac
flac
flac-dbg
flac-dev
flac-doc
flac-src
libflac
libflac++
```

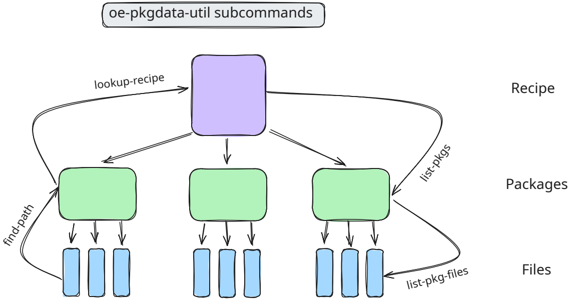
```
$ oe-pkgdata-util list-pkg-files libflac
libflac:
 /usr/lib/libFLAC.so.12
 /usr/lib/libFLAC.so.12.1.0
```

```
$ oe-pkgdata-util find-path /usr/lib/libFLAC.so.12
libflac: /usr/lib/libFLAC.so.12
```

```
$ oe-pkgdata-util lookup-recipe libflac
flac
```

Add Packages to an Image — Key Takeaways

- `IMAGE_INSTALL` takes **package names**, not recipe names
- One recipe can generate multiple packages: binaries, libraries, documentation...
- `oe-pkgdata-util` allows to navigate between recipes, packages and files



Lab — Add Packages to an Image

To add software that we will need in later labs

- Look for recipes
- Look for packages to install

More Yocto Tricks and Tips

Check out our training course:
`https://rootcommit.com/training/yocto/`