

# How to integrate an external software package

Stable: 11.02.2019 - 13:21 / Revision: 09.11.2018 - 15:07

## Contents

1 Article purpose .....	1
2 Using the STM32MPU Embedded Software Developer Package .....	1
3 Using the STM32MPU Embedded Software Distribution Package .....	2
4 References .....	3

## 1 Article purpose

This article provides guidelines to install a software package containing binaries (e.g. tools) not present by default in the OpenSTLinux distribution.

This installation can be done either by using a Yocto recipe in the scope of a STM32MPU Embedded Software Distribution Package or either by cloning an existing tarball or repo git in the scope of a STM32MPU Embedded Software Developer Package.

## 2 Using the STM32MPU Embedded Software Developer Package

Following basic steps to be done :

- Downloading application source code via git clone (tarball or repo git)
- Compiling the application (very often with basic **make** command)
  - Please ensure [SDK environment for compiling Linux application](#) setup is ready
  - Refer to `<Linux kernel installation directory>/README.HOW_TO.txt` helper file to know how to compile (the latest version of this helper file is also available in this user guide: [README.HOW\\_TO.txt](#)).
- Installing the application (very often with basic **make install** command)
  - Refer to application README file to know how to install
- Deploying the application on board

Check where the generated binaries/libraries have been installed and push them onto the board.

```
PC $> scp -r <application_install_folder>/* root@<board ip address>:/<dest_path>
```

## 3 Using the STM32MPU Embedded Software Distribution Package

Following steps to be done:

- Identifying the recipe building and installing the application/package you need
  - More often search this recipe either in layer openembedded-core/meta/ either in layers meta-openembedded/meta-\*/
  - If no existing recipe, you have to create and copy it in your own customer layer: refer to "Writing a New Recipe"<sup>[1]</sup> and [How to create your own distribution](#)
  - Check if the layer containing the recipe is currently in use and so can be compiled, if not you have this error :

```
PC $> bitbake <recipe name>  
ERROR: Nothing PROVIDES '<recipe name>'. Close matches:
```

- Checking if the layer containing the recipe is currently in use

The command to show the layers currently in your build:

```
PC $> bitbake-layers show-layers
```

It outputs a list of the layers currently in use, and their priorities. If a package exists in two or more layers, it will be built from the layer with the higher priority.

**If the recipe is not contained in the list of layers, you must add the layer containing this recipe inside the bblayers.conf.sample of the distro you have selected.**

For instance for distro openstlinux, bblayers file is located here :

```
* meta-st/meta-st-openstlinux/conf/template/bblayers.conf.sample
```

So for instance, to add meta-qt5 layer, just copy/paste this line in bblayers file :

```
* ADDONSLAYERS += "${@}${OEROOT}/meta-qt5' if os.path.isfile('${OEROOT}/meta-qt5/conf/lay
```

- Installing the package for the targetted image, for instance :

```
PC $> echo 'IMAGE_INSTALL_append += "<recipe name>"' >> meta-st/meta-st-openstlinux/recipe
```

- Rebuilding the image:

```
PC $> bitbake st-image-weston
```

- Updating your board with new image binaries. See [Flashing the built image](#).

## 4 References

---

1. [↑ Writing a New Recipe](#)