



How to create your own distribution



Contents

1. How to create your own distribution	3
2. How to create a new open embedded layer	8



A quality version of this page, approved on *19 October 2020*, was based off this revision.

Contents

1 Article purpose	4
2 Prerequisites	5
3 Creating your own distribution	6
3.1 Creating a layer for a new distro	6
3.2 Creating the distribution configuration file	6
3.3 Providing miscellaneous variables	6
3.4 Adding more to the layer if necessary	6
3.5 Use of meta-st-stm32mp with a core image	7
4 Reference list	8



1 Article purpose

The purpose of this article is to describe the basic steps required to create your own distribution.



2 Prerequisites

OpenSTLinux distribution must be installed and into the board Flash memory(ies).



3 Creating your own distribution

As recommended in Yocto user manual ^[1], you may create your own distribution in order not to alter any original distribution Metadata, while gaining more control over package alternative selections, compile-time options, and other low-level configurations.

The basic steps for creating a distribution are detailed in the below chapter.

More details can be found in *Yocto Mega manual/Creating your own distribution*^[1].

3.1 Creating a layer for a new distro

Please read the [How to create a new open embedded layer](#) article.

3.2 Creating the distribution configuration file

Some configuration examples are provided in ST distribution under: `<path of OpenSTLinux distribution delivery>/meta-st/meta-st-openstlinux/conf/distro/*.conf`

3.3 Providing miscellaneous variables

Some miscellaneous variable examples are provided under : `<path of OpenSTLinux distribution delivery>/meta-st/meta-st-openstlinux/conf/distro/include/st-default-distro*.inc` files

All meta-st-openstlinux distro layer configuration files presented above are located here:

```

distro
├── include
│   ├── exception-gplv3.inc
│   ├── openstlinux.inc
│   ├── st-default-distro-providers.inc
│   └── st-default-distro-rules.inc
├── openstlinux-eglfs.conf
├── openstlinux-weston.conf
├── openstlinux-x11.conf
└── [...]

```

3.4 Adding more to the layer if necessary

More add-on component examples:

- recipes for installing distro-specific configuration files
- any image recipes specific to user distribution
- a *psplash append file* for a branded splash screen
- any other append files to make custom changes

Some examples of such add-on components can be found in `<path of OpenSTLinux distribution delivery>/meta-st/meta-st-openstlinux`, you will retrieve some examples of these addons.



ST has already added some recipes (*bbappend) in openstlinux-weston distribution for configuring, patching, ... (non-exhaustive list shown below):

- recipes-benchmark for *glmark2*
- recipes-connectivity for *bluez5*
- recipes-core for *busybox*
- recipes-graphics for *weston-init*

...

Some other added components (*bb) are more specific: images, system services, ... (non-exhaustive list shown below):

- recipes-core for *psplash screen, systemd services*
- recipes-samples for example images
- recipes-security for *OP-TEE userland part*

...

3.5 Use of meta-st-stm32mp with a core image

If you want to use the meta-st-stm32mp layer with a core image (nodistro mode), please apply the following steps to manage the dependencies between layers:

```
PC $> source layers/openembedded-core/oe-init-build-env
PC $> bitbake-layers add-layer ../layers/meta-openembedded/meta-oe
PC $> bitbake-layers add-layer ../layers/meta-openembedded/meta-python
PC $> bitbake-layers add-layer ../layers/meta-st/meta-st-stm32mp
PC $> bitbake core-image-base or bitbake core-image-minimal
```



4 Reference list

- 1.01.1 Yocto Megamanual Creating your own distribution

Flash memories combine high density and cost effectiveness of EPROMs with the electrical erasability of EEPROMs. For this reason, the Flash memory market is one of the most exciting areas of the semiconductor industry today and new applications requiring in system reprogramming, such as cellular telephones, automotive engine management systems, hard disk drives, PC BIOS software for Plug & Play, digital TV, set top boxes, fax and other modems, PC cards and multimedia CD-ROMs, offer the prospect of very high volume demand.

Open Portable Trusted Execution Environment

Stable: 19.10.2020 - 14:17 / Revision: 19.10.2020 - 14:16

Contents

1 Article purpose	9
2 Prerequisites	10
3 Creating your own distribution	11
3.1 Creating a layer for a new distro	11
3.2 Creating the distribution configuration file	11
3.3 Providing miscellaneous variables	11
3.4 Adding more to the layer if necessary	11
3.5 Use of meta-st-stm32mp with a core image	12
4 Reference list	13



1 Article purpose

The purpose of this article is to describe the basic steps required to create your own distribution.



2 Prerequisites

OpenSTLinux distribution must be installed and into the board Flash memory(ies).



3 Creating your own distribution

As recommended in Yocto user manual ^[1], you may create your own distribution in order not to alter any original distribution Metadata, while gaining more control over package alternative selections, compile-time options, and other low-level configurations.

The basic steps for creating a distribution are detailed in the below chapter.

More details can be found in *Yocto Mega manual/Creating your own distribution*^[1].

3.1 Creating a layer for a new distro

Please read the [How to create a new open embedded layer](#) article.

3.2 Creating the distribution configuration file

Some configuration examples are provided in ST distribution under: `<path of OpenSTLinux distribution delivery>/meta-st/meta-st-openstlinux/conf/distro/*.conf`

3.3 Providing miscellaneous variables

Some miscellaneous variable examples are provided under : `<path of OpenSTLinux distribution delivery>/meta-st/meta-st-openstlinux/conf/distro/include/st-default-distro*.inc` files

All meta-st-openstlinux distro layer configuration files presented above are located here:

```

distro
├── include
│   ├── exception-gplv3.inc
│   ├── openstlinux.inc
│   ├── st-default-distro-providers.inc
│   └── st-default-distro-rules.inc
├── openstlinux-eglfs.conf
├── openstlinux-weston.conf
├── openstlinux-x11.conf
└── [...]

```

3.4 Adding more to the layer if necessary

More add-on component examples:

- recipes for installing distro-specific configuration files
- any image recipes specific to user distribution
- a *psplash append file* for a branded splash screen
- any other append files to make custom changes

Some examples of such add-on components can be found in `<path of OpenSTLinux distribution delivery>/meta-st/meta-st-openstlinux`, you will retrieve some examples of these addons.



ST has already added some recipes (*bbappend) in openstlinux-weston distribution for configuring, patching, ... (non-exhaustive list shown below):

- recipes-benchmark for *glmark2*
- recipes-connectivity for *bluez5*
- recipes-core for *busybox*
- recipes-graphics for *weston-init*

...

Some other added components (*bb) are more specific: images, system services, ... (non-exhaustive list shown below):

- recipes-core for *psplash screen*, *systemd services*
- recipes-samples for example images
- recipes-security for *OP-TEE userland part*

...

3.5 Use of meta-st-stm32mp with a core image

If you want to use the meta-st-stm32mp layer with a core image (nodistro mode), please apply the following steps to manage the dependencies between layers:

```
PC $> source layers/openembedded-core/oe-init-build-env
PC $> bitbake-layers add-layer ../layers/meta-openembedded/meta-oe
PC $> bitbake-layers add-layer ../layers/meta-openembedded/meta-python
PC $> bitbake-layers add-layer ../layers/meta-st/meta-st-stm32mp
PC $> bitbake core-image-base or bitbake core-image-minimal
```



4 Reference list

- 1.01.1 Yocto Megamanual Creating your own distribution

Flash memories combine high density and cost effectiveness of EPROMs with the electrical erasability of EEPROMs. For this reason, the Flash memory market is one of the most exciting areas of the semiconductor industry today and new applications requiring in system reprogramming, such as cellular telephones, automotive engine management systems, hard disk drives, PC BIOS software for Plug & Play, digital TV, set top boxes, fax and other modems, PC cards and multimedia CD-ROMs, offer the prospect of very high volume demand.

Open Portable Trusted Execution Environment