Create a New Meta Layer and Write a Hello World Recipe in Yocto Project.

In this tutorial, you will learn how to create a new meta-layer and how to write a new hello world bitbake recipe in the Yocto Project. We will build the image with the hello recipe for QEMU and verify this package present in the rootfs.

1-Create a new meta layer and add it to bblayer.conf file

- Yocto provides a tool that can create the meta-layer and maintain the directory structure which Yocto Project supports.
- Run the below command to create a new layer.

```
$ cd poky
$ source oe-init-build-env
# Now, build folder would your current working director
y.

$ bitbake-layers create-layer ../meta-tutorial
# This command creates the meta-tutorial layer inside th
```

```
e poky directory.

# Add this layer into bblayer.conf file
$ bitbake-layers add-layer ../meta-tutorial

# Display all layer present in the bblayer.conf file.
$ bitbake-layers show-layers
```



```
tutorial@adda:~/yocto/poky/build$
tutorial@adda:~/yocto/poky/build$ bitbake-layers create-layer ../meta-tutorial
NOTE: Starting bitbake server...
Add your new layer with 'bitbake-layers add-layer ../meta-tutorial'
tutorial@adda:~/yocto/poky/build$ bitbake-layers add-layer ../meta-tutorial
NOTE: Starting bitbake server...
tutorial@adda:~/yocto/poky/build$ bitbake-layers show-layers
NOTE: Starting bitbake server...
layer                 path                                      priority
==========================================================================
meta                  /home/tutorial/yocto/poky/meta            5
meta-poky             /home/tutorial/yocto/poky/meta-poky       5
meta-yocto-bsp        /home/tutorial/yocto/poky/meta-yocto-bsp  5
meta-tutorial         /home/tutorial/yocto/poky/meta-tutorial   6
tutorial@adda:~/yocto/poky/build$
```

2-Create Directory For Recipe and Source Files

- Our meta tutorial layer directory structure looks like this.

```
tutorial@adda:~/yocto/poky/meta-tutorial$ tree
.
├── conf
│   └── layer.conf
├── COPYING.MIT
├── README
└── recipes-example
    └── example
        └── example_0.1.bb

3 directories, 4 files
tutorial@adda:~/yocto/poky/meta-tutorial$
```

- We need to create a hello and files directory at the below location.
- **poky/meta-tutorial/recipe-example/hello**
- **poky/meta-tutorial/recipe-example/hello/files/**

3-Write the simple hello world c program

Create the hello.c file at the **poky/meta-tutorial/recipe-example/hello/files/hello.c**

```
//Simple Hello World Program
#include<stdio.h>
int main() {
printf("Hello World , Created Bitbake recipe successfull
```

```
y\n");
return 0;
}
```

4-Write the simple hello recipe file

Create hello_1.0.bb recipe file at the **poky/meta-tutorial/recipe-example/hello/hello_1.0.bb**


```
DESCRIPTION = "Simple helloworld application"
LICENSE = "MIT"
LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/MIT;md5
=0835ade698e0bcf8506ecda2f7b4f302"

SRC_URI = "file://hello.c"

S = "${WORKDIR}"

do_compile() {
        ${CC} hello.c ${LDFLAGS} -o hello
}

do_install() {
        install -d ${D}${bindir}
        install -m 0755 hello ${D}${bindir}
}
```

Reference: Hello recipe

- This hello recipe fetch the source file(hello.c) using the SRC_URI variable and do_compile used to compile the hello.c source file and generated the hello binary.
- do_install function install hello binary at the /usr/bin of the target rootfs.
- Now, the latest directory looks like this.

```
tutorial@adda:~/yocto/poky/meta-tutorial$ tree
.
├── conf
│   └── layer.conf
├── COPYING.MIT
├── README
└── recipes-example
    ├── example
    │   └── example_0.1.bb
    └── hello
        ├── files
        │   └── hello.c
        └── hello_1.0.bb

5 directories, 6 files
tutorial@adda:~/yocto/poky/meta-tutorial$
```

5-Select machine configuration and Add hello package to rootsfs

We are building an image for machine QEMUx86-64 so we have to add this machine in conf/local.conf file.

```
#By default, this machine selection is enabled.
MACHINE ??= "qemux86-64"

#We need to add the hello software package to the target image
IMAGE_INSTALL_append = " hello"
```



```
#
# This sets the default machine to be qemux86-64 if no other machine is selected:
MACHINE ??= "qemux86-64"

#Adding hello package to the image
IMAGE_INSTALL_append = " hello"
```

6-Build Image

Run bitbake to build the minimal boot image for QEMU.

```
$bitbake core-minimal-image
```

```
tutorial@adda:~/yocto/poky/build$ bitbake core-image-minimal
Loading cache: 100% |#########################################################################################
Loaded 1347 entries from dependency cache.
Parsing recipes: 100% |#######################################################################################
Parsing of 783 .bb files complete (781 cached, 2 parsed). 1347 targets, 42 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "1.47.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "universal"
TARGET_SYS           = "x86_64-poky-linux"
MACHINE              = "qemux86-64"
DISTRO               = "poky"
DISTRO_VERSION       = "3.1+snapshot-20201011"
TUNE_FEATURES        = "m64 core2"
TARGET_FPU           = ""
meta
meta-poky
meta-yocto-bsp
meta-tutorial        = "master:c24ece07fa7ded61ac614854a5b9d7ec46e60bd7"

Initialising tasks: 100% |####################################################################################
Sstate summary: Wanted 9 Found 0 Missed 9 Current 1370 (0% match, 99% complete)
NOTE: Executing Tasks
Currently  1 running tasks (3087 of 3720)  82% |##############################################################
0: hello-1.0-r0 do_configure - 0s (pid 10049)
```

7-Run the QEMU image and verify the installed Package

Use the below command to run the QEMU image on Your host PC.

```
$runqemu qemux86-64
```

We installed the hello package at /usr/bin so run hello from the terminal and it gives the output.

```
root@qemux86-64:~#
root@qemux86-64:~#
root@qemux86-64:~# hello
Hello World , Created Bitbake recipe successfully
root@qemux86-64:~#
root@qemux86-64:~# ls /usr/bin | grep hello
hello
root@qemux86-64:~#
```

« Quick Start Your First Yocto Project Build

Create Linux image for Beaglebone Black using Yocto build system »

Add comment

Name (required)

E-mail (required, but will not display)

1000 symbols left

Notify me of follow-up comments

Refresh

Send

JComments

Popular Articles on Yocto

Create your own Linux image for the Raspberry pi board using Yocto Project.

Create a New Meta Layer and Write a New Recipe in Yocto Project

What is the MACHINE_FEATURES and DISTRO_FEATURES variables in the Yocto Project

Quick Start Your First Yocto Project Build

How to do full clean build in Yocto Project

Useful Articles

do_fetch: Fetcher failure Unable to fetch URL from any source

Failed to run qemu: X Error: BadValue in the Yocto Project

Failed to start the bitbake server on Yocto Project

OE-core's config sanity checker detected a potential misconfiguration

Set Number of threads for bitbake in Yocto Project by BB_NUMBER_THREADS