Yocto Project Training

Part 2 - Writing recipes

Fabio Berton

May 11, 2020

Outline

Recipe: Shell script example:

```
DESCRIPTION = "Shell script example"
LICENSE = "MIT"
LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/MIT;md5=0835ade698e0b
↪  cf8506ecda2f7b4f302"

SRC_URI = "file://${BPN}"

S = "${WORKDIR}"

do_install() {
    install -Dm 0755 ${PN} ${D}${bindir}/${PN}
}
```

Directories installation variables: Reference

Variable name	Definition	Typical value
prefix	/usr	/usr
base_prefix	*(empty)*	*(empty)*
exec_prefix	${base_prefix}	*(empty)*
base_bindir	${base_prefix}/bin	/bin
base_sbindir	${base_prefix}/sbin	/sbin
base_libdir	${base_prefix}/lib	/lib
datadir	${prefix}/share	/usr/share
sysconfdir	/etc	/etc
localstatedir	/var	/var
infodir	${datadir}/info	/usr/share/info
mandir	${datadir}/man	/usr/share/man
docdir	${datadir}/doc	/usr/share/doc
servicedir	/srv	/srv
bindir	${exec_prefix}/bin	/usr/bin
sbindir	${exec_prefix}/sbin	/usr/sbin
libexecdir	${exec_prefix}/libexec	/usr/libexec
libdir	${exec_prefix}/lib	/usr/lib

What do I need to know?

- Where is the `sh` file?
- Where are the `S` and `WORKDIR`?
- Why there is only `do_install` task?
- Look at build history!!!
- Inspect all variables with `ye`
- Run *QEMU*

Recipe: C example

```
DESCRIPTION = "Simple C code example"
LICENSE = "MIT"
LIC_FILES_CHKSUM =
↪  "file://COPYING;md5=838c366f69b72c5df05c96dff79b35f2"

SRCREV = "e53d5cd4498c76ca8c58e18d0d14891d20f67eab"
SRC_URI = "git://github.com/fbertux/${BPN}.git;protocol=https"

S = "${WORKDIR}/git"

do_compile() {
    ${CC} ${PN}.c -o ${PN}
}

do_install() {
    install -Dm 0755 ${PN} ${D}${bindir}/${PN}
}
```

What do I need to know?

- Where is the source code?
- Where are the `S` and `WORKDIR`?
- How to compile and install *C* code
- Look at build history!!!
- Inspect all variables with `ye`
- Run *QEMU*

There's more:

- Others variables needed to build a *C* code:

```
${CFLAGS}
${LDFLAGS}
```

Recipe: Makefile example

```
DESCRIPTION = "Makefile example"
LICENSE = "MIT"
LIC_FILES_CHKSUM =
↪  "file://COPYING;md5=838c366f69b72c5df05c96dff79b35f2"

SRCREV = "2c0f9eacec6c886991563f092519a55e1c74dc92"
SRC_URI = "git://github.com/fbertux/${BPN}.git;protocol=https"

S = "${WORKDIR}/git"

do_configure[noexec] = "1"

do_compile() {
    oe_runmake
}

do_install() {
    oe_runmake DESTDIR=${D} install
}
```

What do I need to know?

- How to compile and install Makefile based projects
- What is `oe_runmake`?
- Look at build history!!!
- Inspect all variables with `ye`
- Run *QEMU*

Recipe: Autotools example

```
DESCRIPTION = "Autotools example"
LICENSE = "MIT"
LIC_FILES_CHKSUM =
↪  "file://COPYING;md5=838c366f69b72c5df05c96dff79b35f2"

SRCREV = "1ee6714b88df91dd44c9aae6d9264a7e7325ce3e"
SRC_URI = "git://github.com/fbertux/${BPN}.git;protocol=https"

S = "${WORKDIR}/git"

inherit autotools
```

What do I need to know?

- How the task are running?
- Look at *autotools* class
- Look at build history!!!
- Inspect all variables with `ye`
- Run *QEMU*

How depends works

- Why does a package need a dependency?
- Where the build system find the dependencies?
- Looking at *sysroot*

Recipe: SDL example

```
DESCRIPTION = "SDL example"
LICENSE = "MIT"
LIC_FILES_CHKSUM =
↪  "file://COPYING;md5=838c366f69b72c5df05c96dff79b35f2"

DEPENDS = "libsdl2"

SRCREV = "4610fe3841f99d8c24d9417f585cd2df0544fe72"
SRC_URI = "git://github.com/fbertux/${BPN}.git;protocol=https"

S = "${WORKDIR}/git"

inherit pkgconfig

do_compile() {
    oe_runmake
}

do_install() {
    install -Dm 0755 ${S}/main ${D}${bindir}/${PN}
```

What do I need to know?

- Why do I need to use `DEPENDS`?
- Where are the includes and *libs* file?
- Inspect *recipes-sysroot*
- Look at build history!!!
- Inspect all variables with `ye`
- Run `QEMU`

Why does a package need a runtime dependency?

Where the package find the dependencies?

Example: `curl` example