In Yocto, how to include header files from another recipes

Asked 4 years, 10 months ago Modified 8 months ago Viewed 10k times

7

My program depends on the `poco` recipes, which provides both the header files and shared libraries. However, I cannot make use of the header files from `poco` in my recipe, which leads to the error `Poco/Delegate.h: No such file for directory`.

How do I make the header available at build time for my software package?

Here is an example recipe:

```
SUMMARY = ""
DESCRIPTION = ""
AUTHOR = ""
LICENSE = "CLOSED"
LIC_FILES_CHKSUM = ""
HOMEPAGE = ""
BUGTRACKER = ""

S = "${WORKDIR}"

SRC_URI = " file://foo.cpp \
            file://CMakeLists.txt \
"

inherit pkgconfig cmake

DEPENDS_foo = "poco"
RDEPENDS_foo = "poco"

do_install() {
    install -d ${D}/${bindir}
    install -m 755 ${S}/foo ${D}/${bindir}
}

FILES_${PN} = "${bindir}/foo"
```

yocto

Share Improve this question Follow

asked Apr 26, 2018 at 5:02
Amumu
17.6k ● 29 ● 84 ● 130

1 basically that look correct, assuming that your recipe is called foo_1.0.bb or such. the DEPENDS doesn't need the package identifier, so maybe its already enough to change DEPENDS_foo to DEPENDS, and of

3 Answers

Sorted by:

Highest score (default)

3

We can use **provider** and **user** to illustrate this case, the package (recipe) provides a header file to be used by another package (recipe) is the **provider**, the package (recipe) use a header file from another package (recipe) is the **user**.

First we change the provider's recipe (**myprovider.bb**) to export the header file -- **myapi.h**,

```
...
do_install() {

    install -d ${D}/${bindir}
    install -m 755 ${B}/hello_provider ${D}/${bindir}

    install -d ${D}${libdir}/lib_myprovider/
    install -m 0755 ${WORKDIR}/myapi.h ${STAGING_DIR_TARGET}${libdir}/lib_myprovider/
}
...
```

Secondly we change the user's recipe (**myuser.bb**) to refer the header file -- **myapi.h**

```
...
do_compile () {

    ${CC} ${WORKDIR}/main.c -o hello_user ${CFLAGS} ${LDFLAGS} -I${STAGING_DIR_TARGET}/${libdir}/lib_myprovider/
}

# file dependency declaration
FILES_${PN} = "${libdir}/lib_myprovider"

# package dependency declaration
DEPENDS += "myprovider"
...
```

At last, rebuild myprovider.bb and myuser.bb recipes, it should work.

Share Improve this answer Follow

edited Apr 2, 2019 at 3:00

answered Mar 20, 2019 at 4:08

Zephyr
6,043 ● 33 ● 33

but where STAGING_DIR_TARGET is defined ? – Mouin Aug 20, 2021 at 17:35

Could we use STAGING_INCDIR instead? Also, could the do_populate_sysroots be used instead of do_install? – Maverickgugu Jun 29, 2022 at 10:55

3

> Recipes should never populate the sysroot directly:

Recommended way is (poco recipe should do something similar):

> Files should be installed into standard locations:

```
...
do_install() {
...
    install -d ${D}${includedir}
    install -m 0755 ${S}/myapi.h ${D}${includedir}/
...
}
...
```

Than, include **poco recipe** as a build dependency of **foo.bb** :

DEPENDS += "poco"

And compile normally.

Share Improve this answer Follow

edited Jul 9, 2019 at 21:45
Khem
1,152 ●8 ●8

answered Jul 9, 2019 at 16:48
fabatera
305 ●2 ●9

0

You could use `DEPENDS` to have dependency with "poco" recipe and it would build and populate "poco" recipe's headers and libs into your recipe's sysroot.

Similar way you have to mention the export paths in the provider's recipe with `FILES_*( * - package type)`

Share Improve this answer Follow

edited Jul 4, 2022 at 13:49

answered Jul 4, 2022 at 5:16
Gowrishankar Saminathan
1 ●2