[BMC] OpenBMC Development Basics 4: Web Development Basics

jiangwei0512 Posted on 2023-08-18 00:05:05 Read 4k Collection 7 Likes 5

Category Column: BMC Article Tags: openbmc

摘要 This article describes in detail how to port open source web code (such as vueui) to the OpenBMC project, and introduces the basic steps of web development using Node.js, npm, and Vue2, including dependency management, build, and deployment processes.

The summary is generated in C Know , supported by DeepSeek-R1 full version, go to experience>

Web Development Basics

We have already introduced how to add custom programs and patch open source Web code. However, if you want to do custom Web development later, the best way is to put the code directly into your own project. This article first completes this step and then briefly introduces the basics of Web development.

Web Code Porting

The recipe for the Web used by the open source OpenBMC is meta-phosphor/recipes-phosphor/webui/webui-vue_git.bb. The corresponding source code can be downloaded from git ://github.com/openbmc/webui-vue.git. Since webui has been used normally before, the code can be found directly in build/downloads. We put the source code here into our own project. Of course, it cannot be used directly. The corresponding bb file needs to be modified. Here it is renamed to vueui.bb. The final directory is as follows:

bash AI generated projects 登录复制

```
jw@HOME:~/openbmc$ tree meta-beni/ -L 3
meta-beni/
├── COPYING.MIT
├── README
├── conf
│   ├── 略
├── recipes-example
│   ├── 略
│   └── vueui
│       ├── vueui
│       └── vueui.bb
└── recipes-phosphor
    ├── 略

16 directories, 11 files
```

收起 ^

The vueui directory contains the source code of webui, and vueui.bb is a modified version of webui-vue_git.bb. Its contents are as follows (the specific modification points have been noted):

bash AI generated projects 登录复制

```
LICENSE = "MIT"
LIC_FILES_CHKSUM = "file://${COREBASE}/meta/COPYING.MIT;md5=3da9cfbcb788c80a0384361b4de20420"
DEPENDS:prepend = "nodejs-native "
PV = "1.0.0"
# This recipe requires online access to build, as it uses NPM for dependency
# management and resolution.
PR = "r1"

# 使用本地的代码，位于vueui目录下
# SRC_URI = "git://github.com/openbmc/webui-vue.git;branch=master;protocol=https"
SRC_URI = "file://public/ \
           file://src/ \
           file://.eslintrc.js \
           file://babel.config.js \
           file://jest.config.js \
           file://package-lock.json \
           file://package.json \
           file://postcss.config.js \
           file://vue.config.js \
"
twen
twen S = "${WORKDIR}"
twen
twen inherit allarch

EXTRA_OENPM ?= ""

export CXX = "${BUILD_CXX}"
export CC = "${BUILD_CC}"
export CFLAGS = "${BUILD_CFLAGS}"
export CPPFLAGS = "${BUILD_CPPFLAGS}"
export CXXFLAGS = "${BUILD_CXXFLAGS}"

# Workaround
# Network access from task are disabled by default on Yocto 3.5
# https://git.yoctoproject.org/poky/tree/documentation/migration-guides/migration-3.5.rst#n25
do_compile[network] = "1"
do_compile () {
    cd ${S}
    # node_modules存放的是Web开发使用到的依赖库，不用每次都删除再重新下载
    # rm -rf node_modules
    # 需要加上--legacy-peer-deps，否则会报错
    # 实际上并不需要每次都install，只需要install一次就可以了，但是关系不大
    # npm --loglevel info --proxy=${http_proxy} --https-proxy=${https_proxy} install
    npm --loglevel info --proxy=${http_proxy} --https-proxy=${https_proxy} install --legacy-peer-deps
    npm run build ${EXTRA_OENPM}
}
do_install () {
   # create directory structure
   install -d ${D}${datadir}/www
   cp -r ${S}/dist/** ${D}${datadir}/www
   find ${D}${datadir}/www -type f -exec chmod a=r,u+w '{}' +
   find ${D}${datadir}/www -type d -exec chmod a=rx,u+w '{}' +
}

FILES:${PN} += "${datadir}/www/*"
```

收起 ^

At this point, you can see the newly added vueui through bitbake:

```bash
jw@HOME:~/openbmc/build$ bitbake -s | grep vueui
vueui                                           :1.0.0-r1
```

Then compile the module using bitbake:

```bash
jw@HOME:~/openbmc/build$ bitbake vueui
Loading cache: 100% |########################################################################################################| Time: 0:00:00
Loaded 4387 entries from dependency cache.
Parsing recipes: 100% |######################################################################################################| Time: 0:00:00
Parsing of 2711 .bb files complete (2710 cached, 1 parsed). 4386 targets, 566 skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies

Build Configuration:
BB_VERSION           = "2.4.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "ubuntu-20.04"
TARGET_SYS           = "arm-openbmc-linux-gnueabi"
MACHINE              = "beni-ast2500"
DISTRO               = "openbmc-phosphor"
DISTRO_VERSION       = "nodistro.0"
TUNE_FEATURES        = "arm thumb arm1176jzs"
TARGET_FPU           = "soft"
meta
meta-poky
meta-oe
meta-networking
meta-python
meta-phosphor
meta-aspeed          = "HEAD:67c9d4e715c705cd05fd04f7c8cd4fad300a4666"
meta-beni            = "master:ce757c7232d1ffcf0d74155bc283aba0591325ce"
workspace            = "HEAD:67c9d4e715c705cd05fd04f7c8cd4fad300a4666"

Initialising tasks: 100% |###################################################################################################| Time: 0:00:00
Sstate summary: Wanted 1 Local 0 Mirrors 0 Missed 1 Current 104 (0% match, 99% complete)
Removing 1 stale sstate objects for arch allarch: 100% |#####################################################################################################| Time: 0:00:00
NOTE: Executing Tasks
NOTE: Tasks Summary: Attempted 460 tasks of which 458 didn't need to be rerun and all succeeded.
```

收起 ^

This means that the local code version of Web is ready for use. You can then build the BMC binary and debug the Web in it. However, you need to replace the Web content in meta-beni/recipes-phosphor/packagegroups/packagegroup-obmc-apps.bbappend before doing so:

```bash
RDEPENDS:${PN}-extras:append = "helloworld memtester vueui" # webui-vue改成vueui
```

Vue version web development basics

The following introduces the development basics of the Vue version of Web.

OpenBMC originally used a Web based on the AngularJS framework (the current version can still find its recipe meta-phosphor/recipes-phosphor/webui/phosphor-webui_git.bb), but since the framework has stopped maintenance, OpenBMC has switched to the Vue framework and uses the Vue2 version.

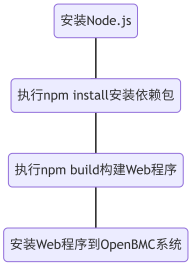
The construction of the Vue version of Web depends on Node.js, so there is the following code in the recipe:

```bash
DEPENDS:prepend = "nodejs-native "
```

Node.js is an open source and cross-platform JavaScript runtime platform, which includes another tool NPM. NPM stands for **Node** Package **Manager ,** which is a package management tool. In building the Vue version Web, its functions include:

- Download the current web dependency package.
- Build the final web application from the web source code.

Corresponding to the recipe, the process is as follows:



1. Download the code of the dependent package:

```bash
npm --loglevel info --proxy=${http_proxy} --https-proxy=${https_proxy} install
```

After executing this command, the download will be based on the content of meta-beni/recipes-example/vueui/vueui/package.json.

It should be noted here that Web development relies on many ready-made libraries, which can be used to simplify development during the development process. If any library is needed, just specify the library name and version information in package.json.

2. Code to build the web application:

```bash
npm run build ${EXTRA_OENPM}
```

The source of this command is also the package.json file, and it ultimately relies on the Vue scaffolding to perform the build operation:

```json
    "scripts": {
        "build": "check-node-version --package && vue-cli-service build",
        // 其它略
    },
```

The generated web program is located in the dist directory, and its entry is an index.html file.

3. Once the program is generated, place it in a fixed location on the OpenBMC system:

bash AI generated projects 登录复制

```
install -d ${D}${datadir}/www
cp -r ${S}/dist/** ${D}${datadir}/www
```

This location is actually determined by the Web server (OpenBMC uses bmcweb as the server, corresponding to the recipe meta-phosphor/recipes-phosphor/interfaces/bmcweb_git.bb), so the two need to be consistent.

The above is a simple process of Web development, which involves the use of Node.js and npm, the basics of the Vue framework, and other knowledge. You need to understand these contents before you can carry out subsequent development.