

# FlowAPP 编译说明书



青岛数智船海科技有限公司

时间	修改纲要	修改人
2024/09/19	首次编制	李宝君



# 1. 编译环境依赖

系统：Ubuntu 20.04

## (1) GCC

- 安装版本：gcc 9
- 安装命令：

<code>sudo apt update</code>
<code>sudo apt install build-essential</code>
<code>sudo apt install gcc-9 g++-9</code>
<code>sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-9 60 --slave /usr/bin/g++ g++ /usr/bin/g++-9</code>

## (2) OpenGL

- 安装版本：OpenGL 2.1 及以上
- 安装命令：

<code>sudo apt install libgl1-mesa-dev</code>
<code>sudo apt install libglu1-mesa-dev</code>
<code>sudo apt install freeglut3 freeglut3-dev</code>
<code>sudo apt install libfreeimage3</code>

## (3) OpenFOAM:

- 安装版本：安装 OpenFOAM11
- 安装命令：

添加官方 apt 源 (由于服务器在国外， 下载速度较慢，建议使用 下方的第三方源下载)	<code>sudo sh -c "wget -O - https://dl.openfoam.org/gpg.key &gt; /etc/apt/trusted.gpg.d/openfoam.asc"</code>
使用第三方源 (★推荐使用)	<code>sudo add-apt-repository http://dl.cfdem.cn/ubuntu</code> <code>sudo add-apt-repository http://dl.cfdem.cn/deb</code>
更新 apt 软件包列表	<code>sudo apt update</code>

安装 OpenFOAM11	<code>sudo apt -y install openfoam11</code>
---------------	---

- 安装参考网址: <https://openfoam.org/download/11-ubuntu/>

#### (4) Qt

- 安装版本: Qt5.14.2
- 安装方式:
- `wget https://download.qt.io/archive/qt/5.14/5.14.2/qt-opensource-linux-x64-5.14.2.run`
- 打开 <https://download.qt.io/archive/qt/5.14/5.14.2/>, 下载 `qt-opensource-linux-x64-5.14.2.run` 文件, 在将其安装到系统上。
- Qt 安装部分问题

1. 提示没有 无法显示 “qt-xxx.run”, 没有安装处理 “executable” 文件的程序

解决方法



① 右键 单击 “qt-opensource-linux-x64-5.14.2.run”

② 左键的单击 “属性”



① ② ③ 将所有 权限设置为 “读取与写入”

④ 勾选 “允许文件作为程序执行”

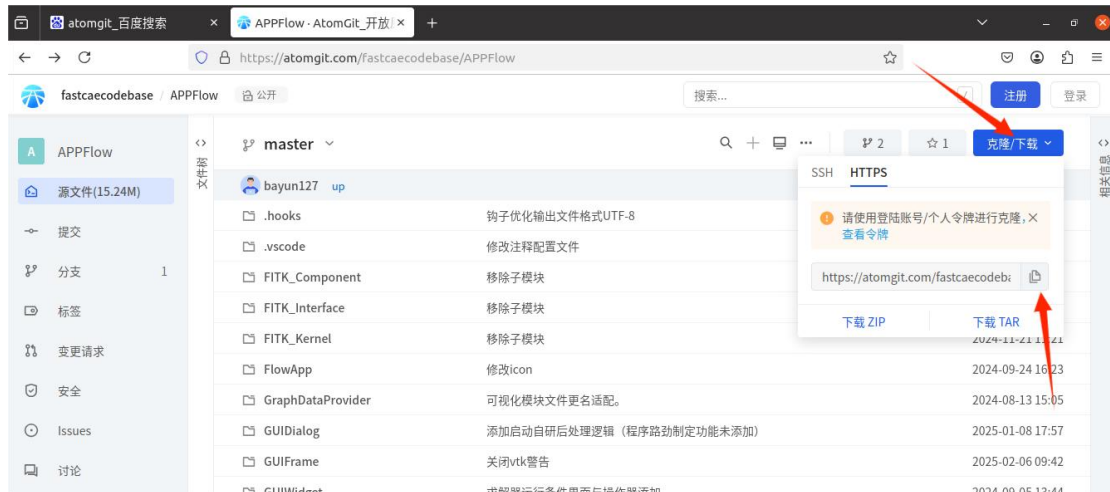
Fast CAE

## 2. 安装包下载

(1) 克隆 APPFlow 项目

➤ 安装命令: (在自定义目录下)

➤ 从 <https://atomgit.com/fastcaecodebase/APPFlow> 界面获取复制链接



```
git clone https://atomgit.com/fastcaecodebase/APPFlow.git
```

若提示 git 命令未识别，需要安装 git 工具，命令如下：

```
sudo apt install git
```

(2) 拉取子仓库 (在 AppFlow 目录下)

➤ 安装命令: (在自定义目录下)

```
chmod +x InitSubmodules.sh (赋予执行权限)
```

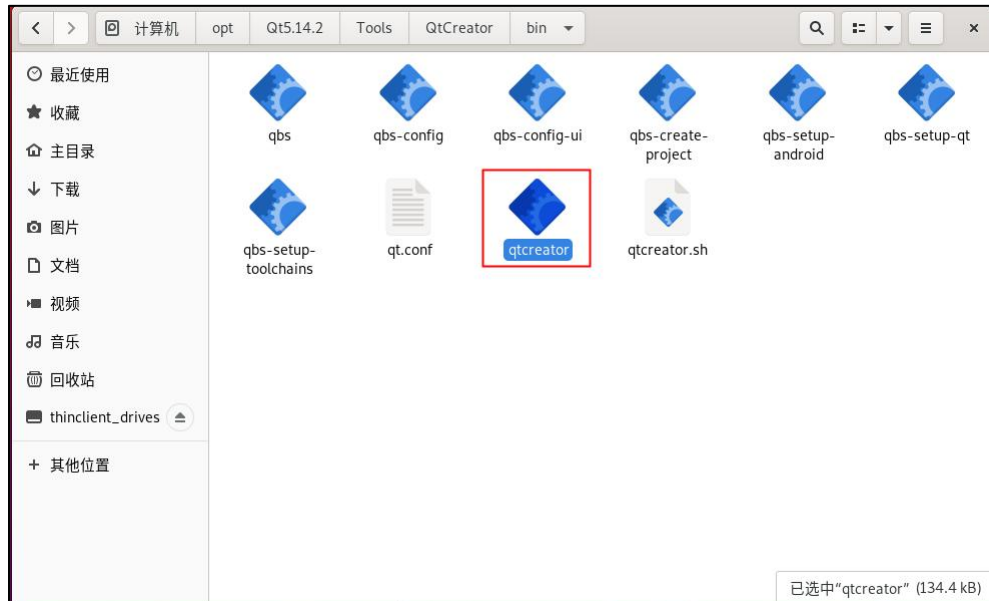
```
bash InitSubmodules.sh (使用 ./InitSubmodules.sh 会提示权限不足)
```

如下图所示

```
root@bayun:/tools/APPFlow# chmod -x InitSubmodules.sh
root@bayun:/tools/APPFlow# bash InitSubmodules.sh
子模组 'FITK_Component/FITKCompMessageWidget' (https://atomgit.com/fastcaecodebase/FITKCompMessageWidget.git) 已对路径 'FITK_Component/FITKCompMessageWidget' 注册
子模组 'FITK_Component/FITKFlowOFIOHDF5' (https://atomgit.com/fastcaecodebase/FITKFlowOFIOHDF5.git) 已对路径 'FITK_Component/FITKFlowOFIOHDF5' 注册
子模组 'FITK_Component/FITKFlowOFIOHDF5' (https://atomgit.com/fastcaecodebase/FITKFlowOFIOHDF5.git) 已对路径 'FITK_Component/FITKFlowOFIOHDF5' 注册
```

### 3. 启动编译

#### 2.1 启动 Qt5.14.2



- ① 左键双击 “qtcreeator”

#### 2.2 选择 FlowApp.pro 文件

完成“2.1 打开 qt”后进入“qt”主界面，选择“欢迎”选项卡 加载“Flowapp.pro”

具体操作如下图所示



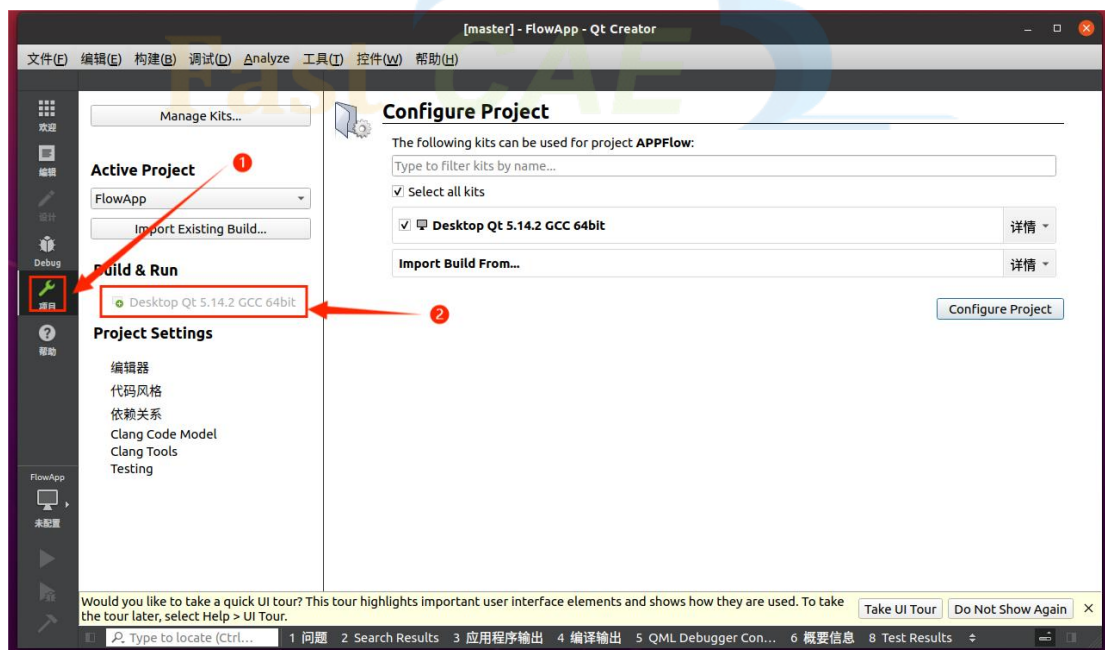
- ① 左键 单击 矩形区域 “欢迎选项卡”
- ② 左键 单击 “Project” 按钮
- ③ 左键 单击 “Open” 按钮



- ① 左键双击 “FlowApp.pro” 文件 后即可加载 “APPFlow” 项目

## 2.3 进入“构建设置界面”

双击 FlowApp.pro 后会自动进入“项目”界面，如无自动跳转，可在左侧 自行选择“项目”  
具体步骤如下图所示



- ① 左键 单击 左侧导航栏 “项目”

② 左键 单击 矩形区域 “Desktop Qt 5.14.2 GCC 64bit”，完成后即可看见“构建设置界面”

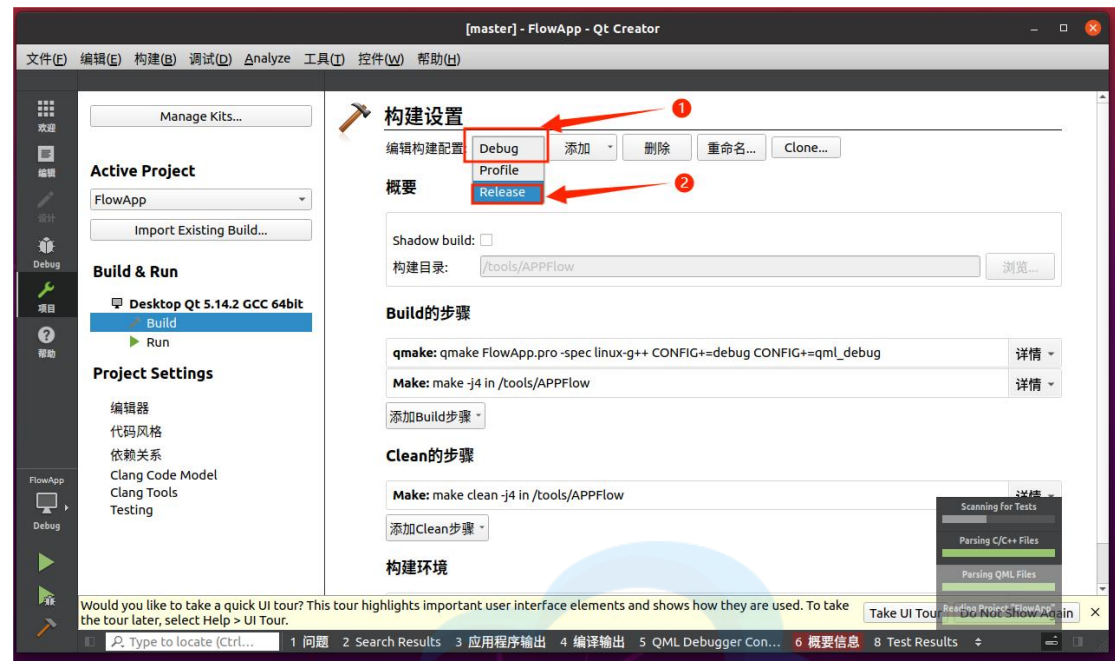


## 2.4 修改模式为“Release”

完成“2.3 进入构建设置界面后”后 在 “构建设置”界面 将“Debug”模式修改为“Release”

具体步骤 如下图所示

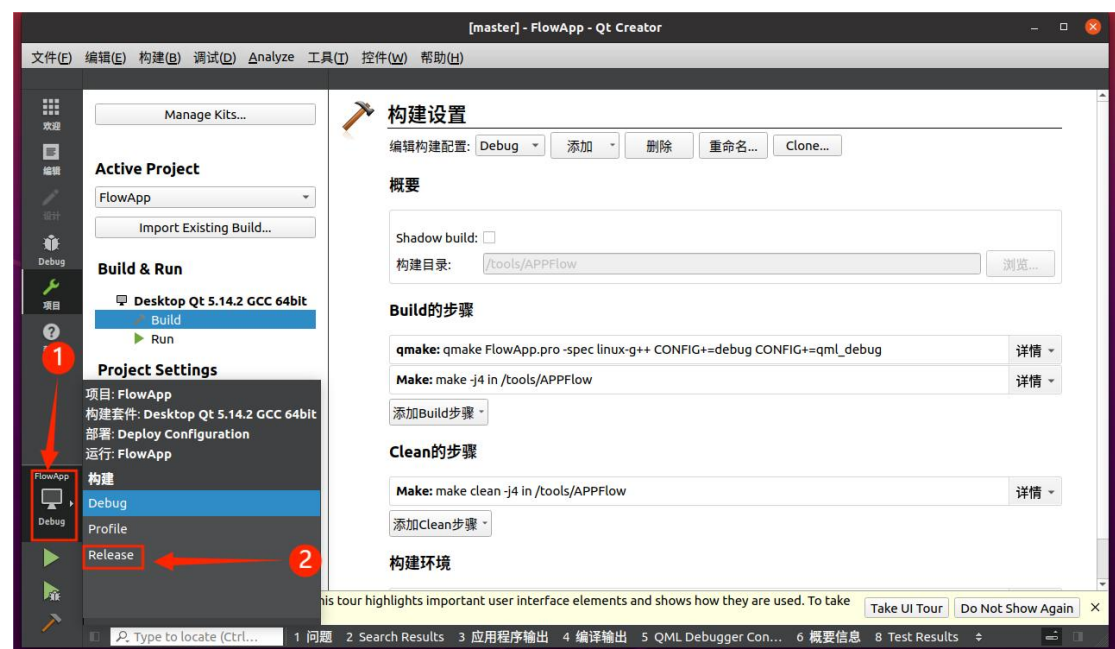
方式一



① 左键 单击 矩形区域 “Debug”

② 左键 单击 “Release”，点击“Release”后 左键单击 界面空白 地方

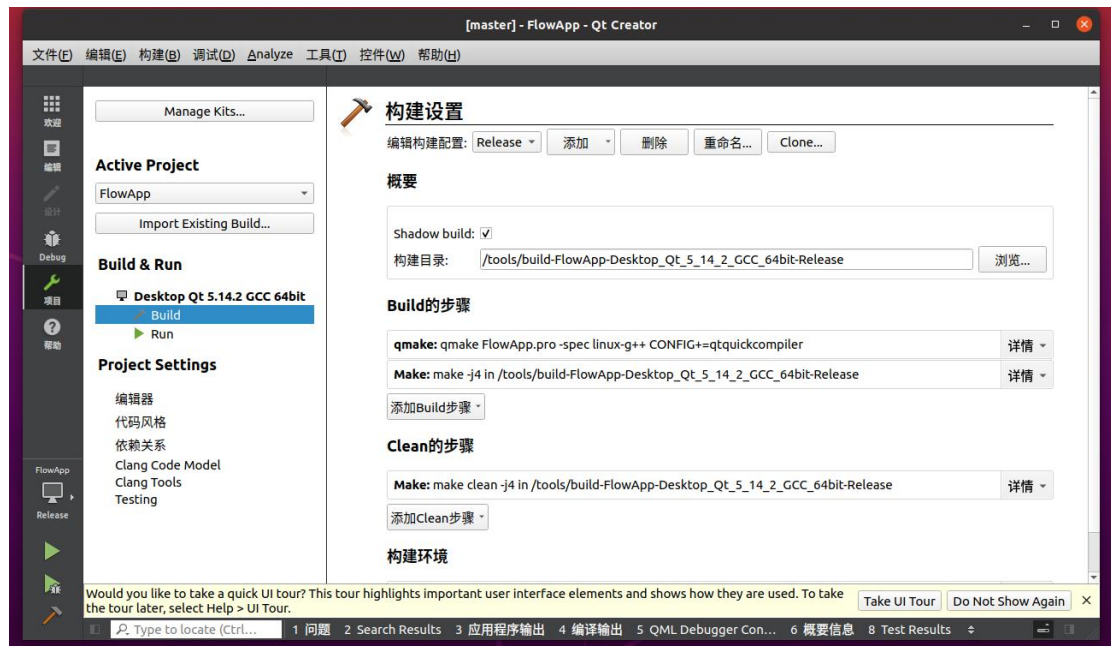
方式二：



① 左键 单击 矩形区域

② 左键 单击 “Release”

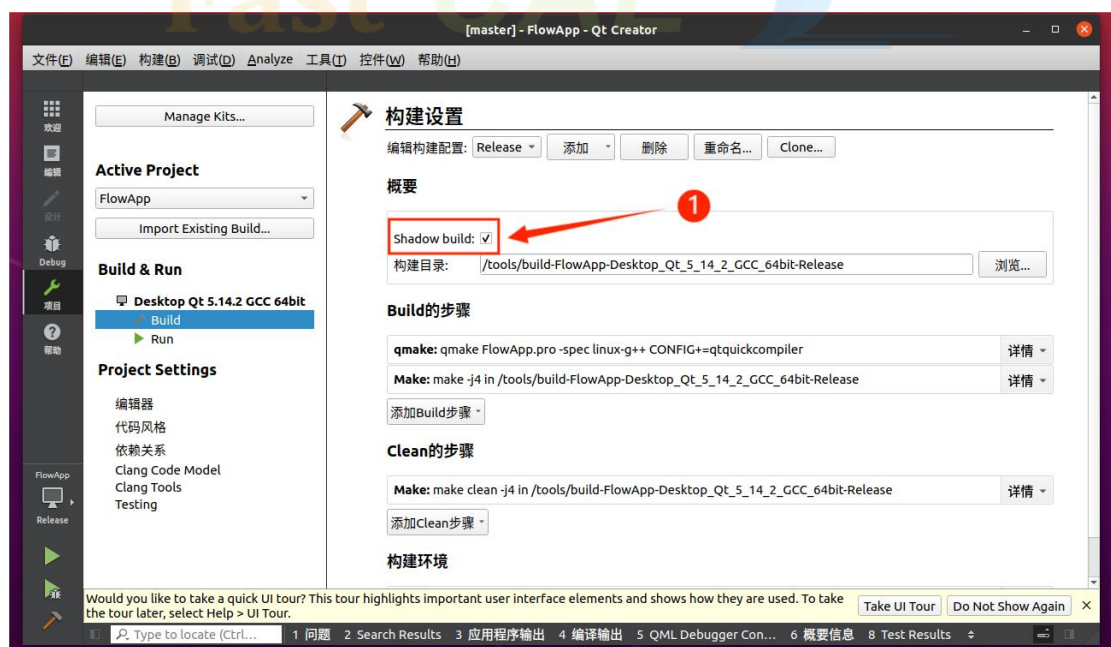
完成后效果如下所示



## 2.4 设置编辑输出路径

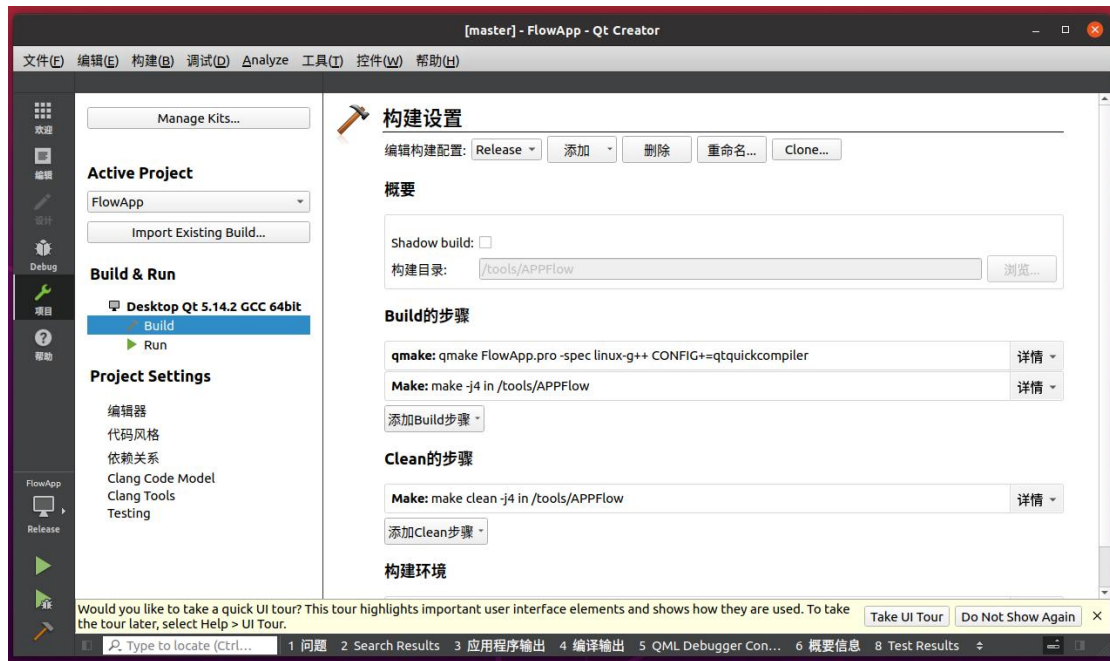
完成“2.3 修改模式为“Release””操作后，取消勾选“Shadow build”。

具体步骤 如下图所示



① 在“构建设置”界面 左键 单击 “Shadow build” 后面的选择框，取消其勾选

完成后效果 如下图所示

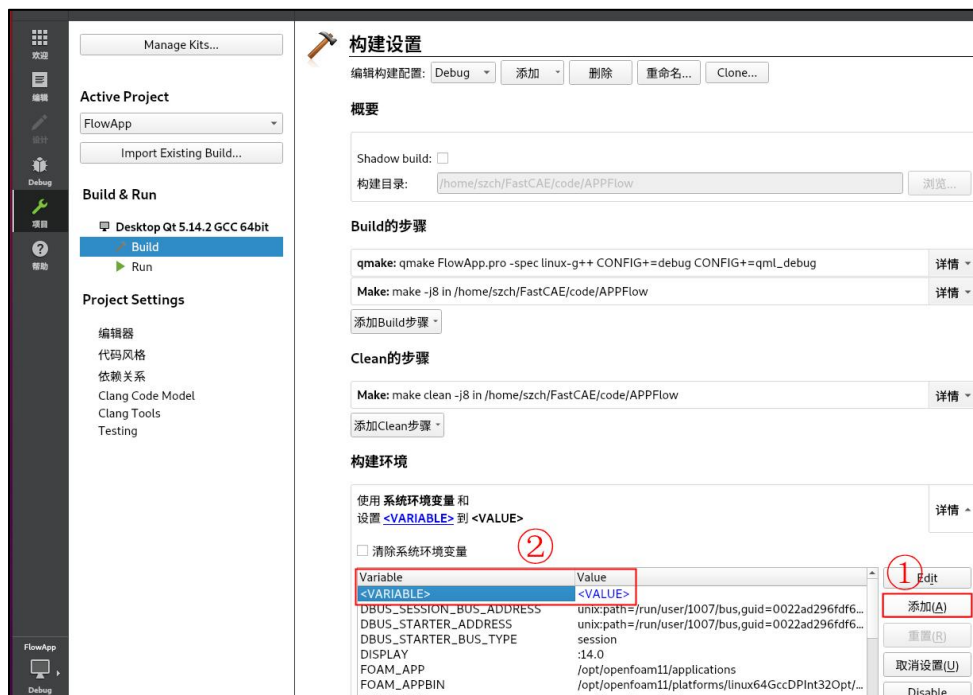


## 2.4 添加环境变量

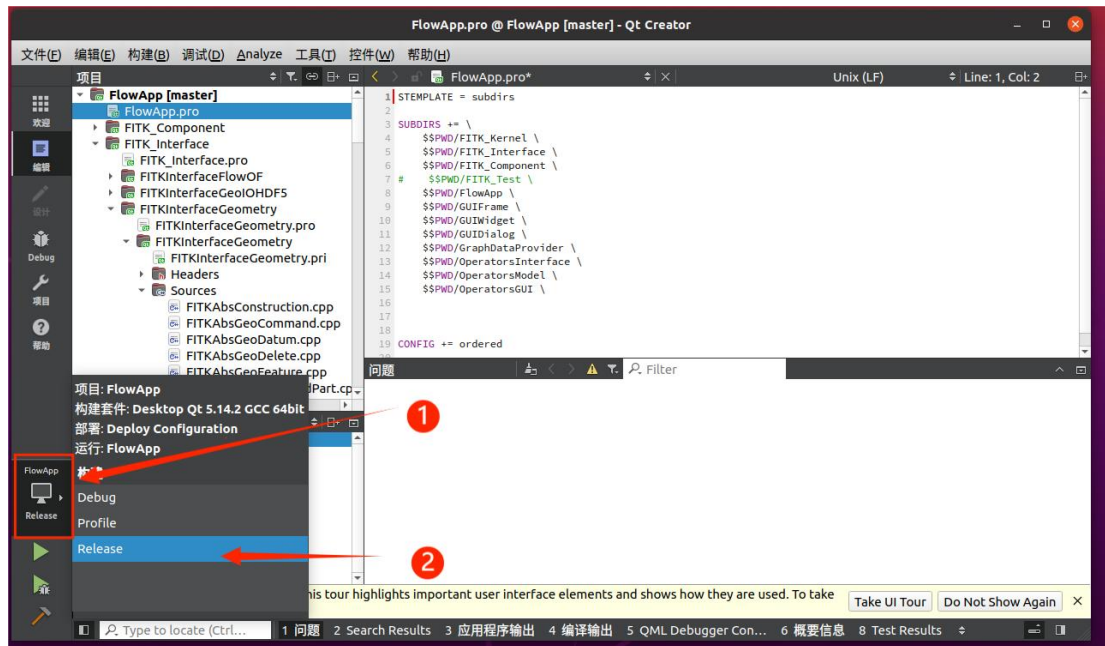
如果系统 OpenGL 采用 Mesa, 需在系统环境变量或 Qt 编译环境变量中, 添加下述命令至系统环境变量或 Qt 编译环境变量中。下图以添加 Qt 编译环境变量为例。

MESA\_GL\_VERSION\_OVERRIDE=4.5

MESA\_GLSL\_VERSION\_OVERRIDE=450



## 2.5 设置构建模式为 release



① 左键单击 矩形区域

② 点击 Release

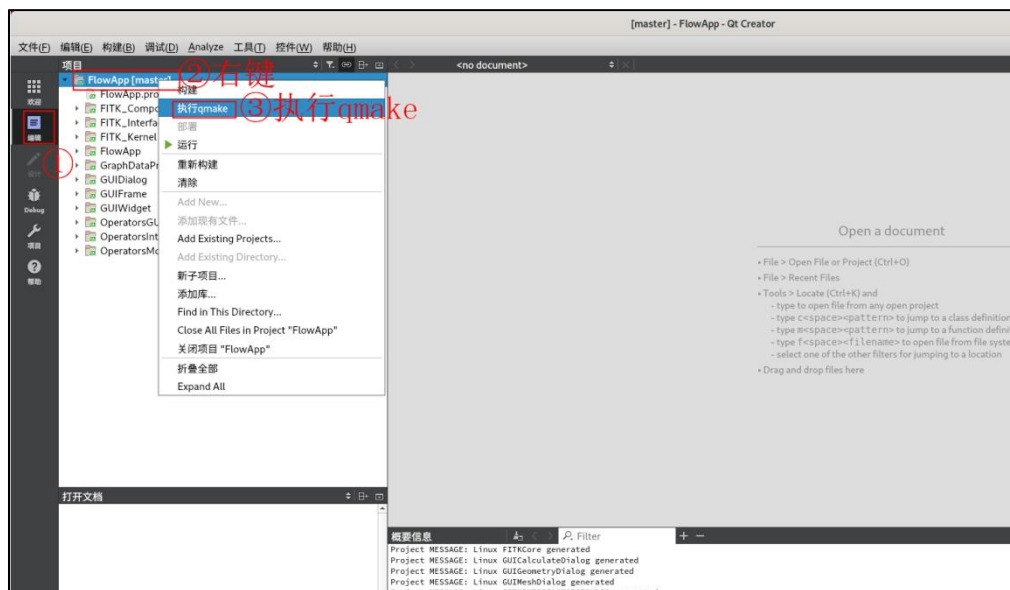
## 2.5 执行 qmake

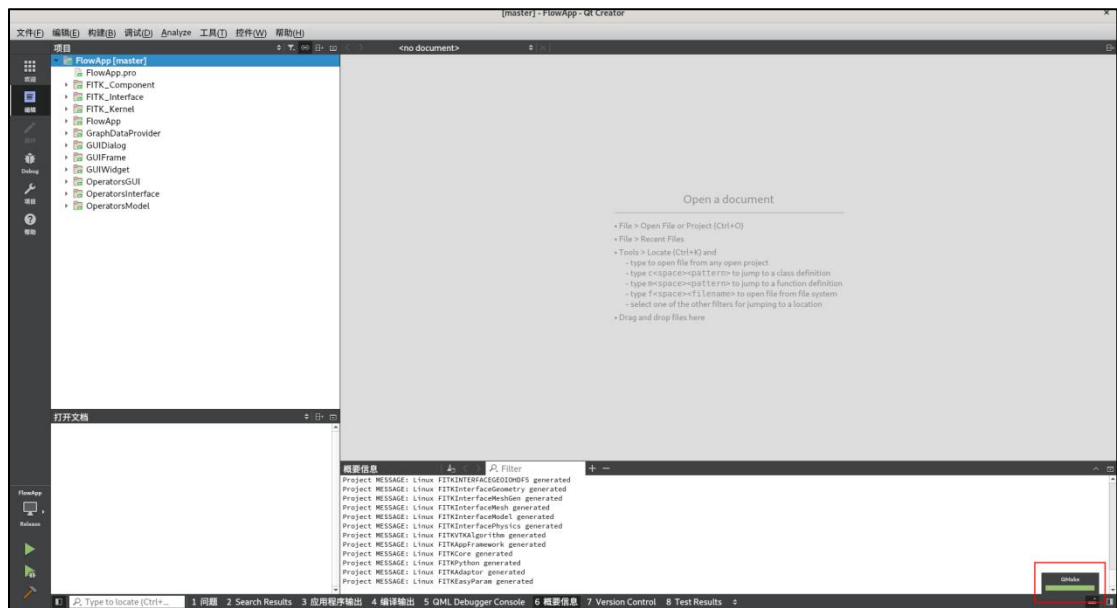
① 点击“编辑”按钮；

② 右键“FlowApp”；

③ 点击“执行 qmake”；

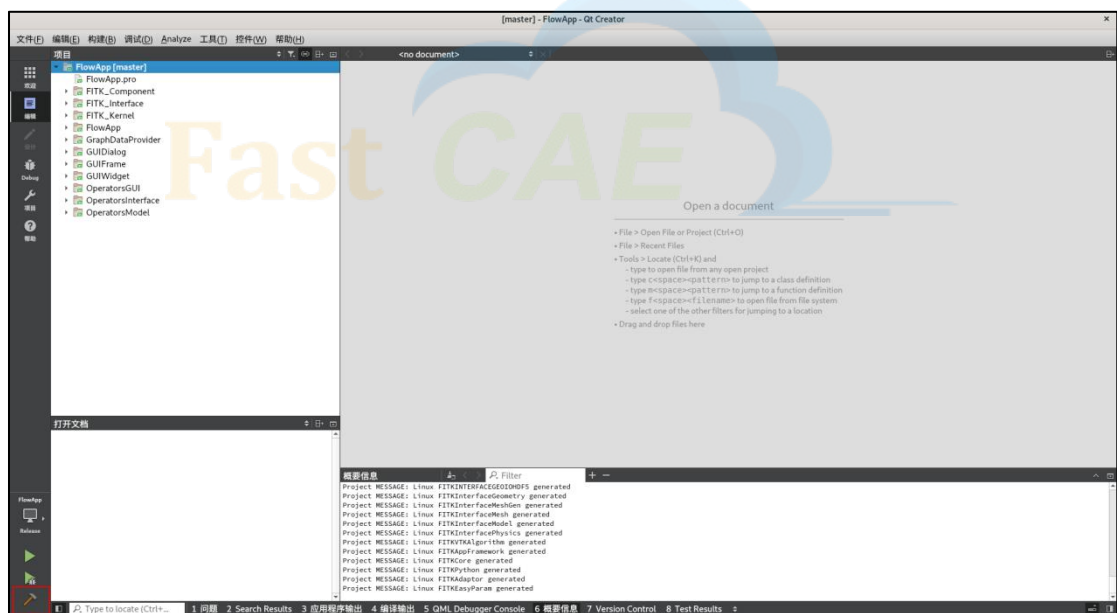
qmake 执行成功后，在程序右下角提示执行成功。





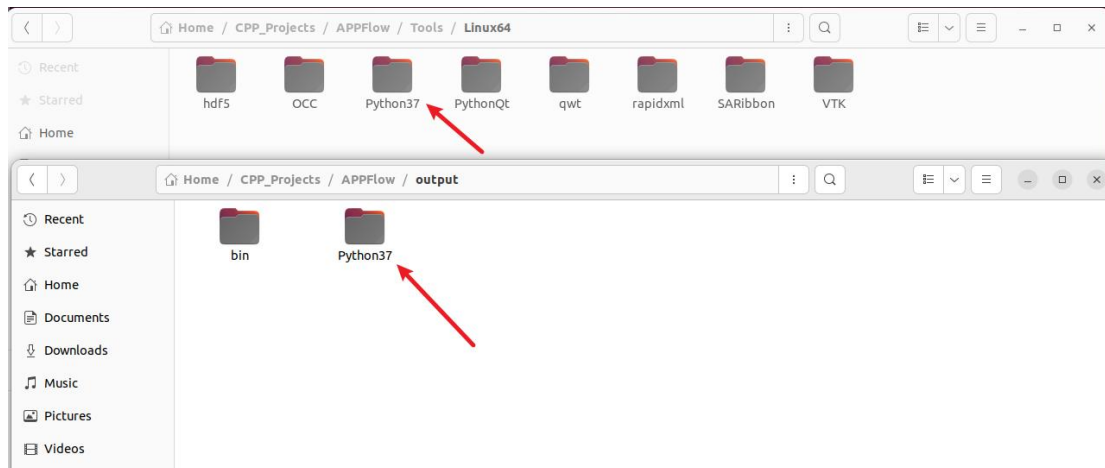
## 2.6 执行编译

qmake 执行成功后，点击左下角编译按钮进行编译。



注意：该程序会依赖 Python 库，需要将项目文件/Tools/Linux64/Python37 该文件夹拷贝到项目文件/output/里面。

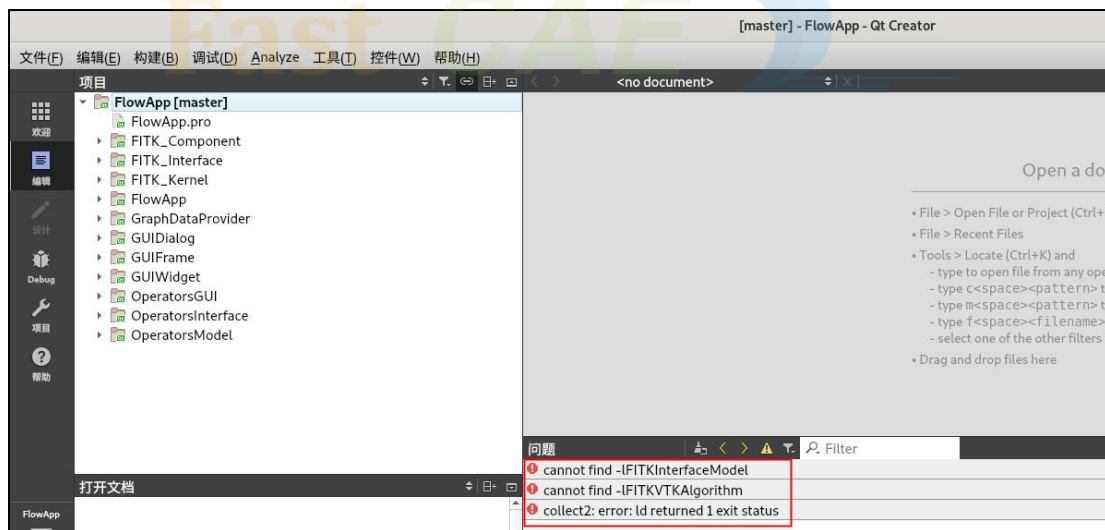




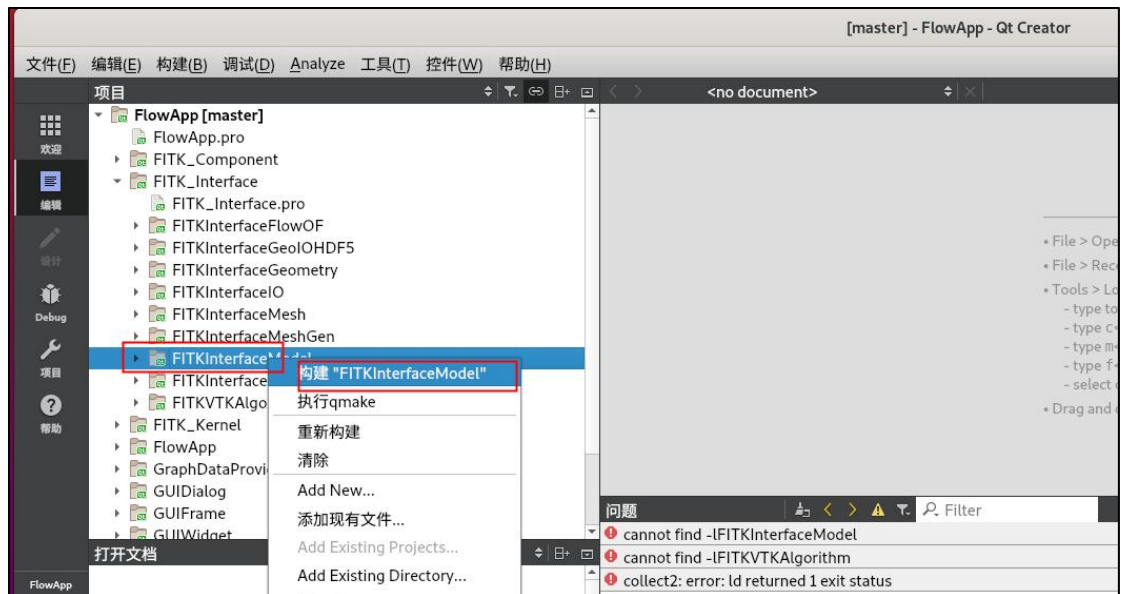
## 2.7 编译过程报误解决

编译过程中会出现两种错误：一种是编译过程中部分项目未编译会导致依赖该项目的其他项目无法编译，需多次编译或手动编译依赖项目（在所依赖的项目节点上右键点击，选择构建）；另一种是编译过程中会出现未依赖外部库错误，需将所需的外部库拷贝至项目目录中。

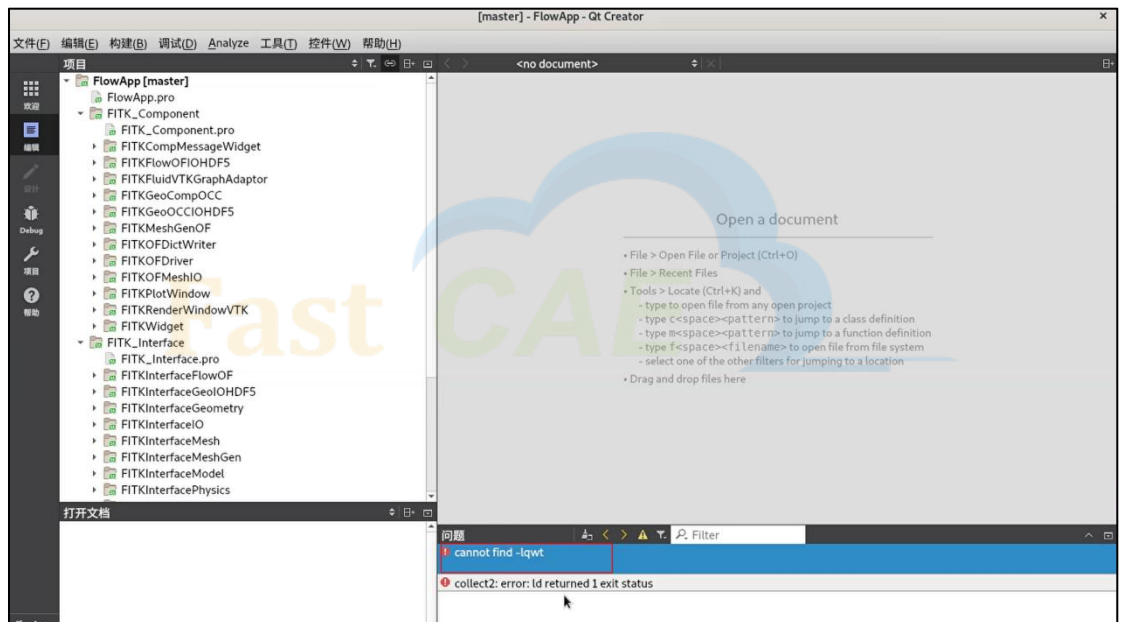
错误 1：被依赖项未编译，如下图所示编译时 FITKInterfaceModel 项目未编译导致依赖该项目的其他项目编译失败。



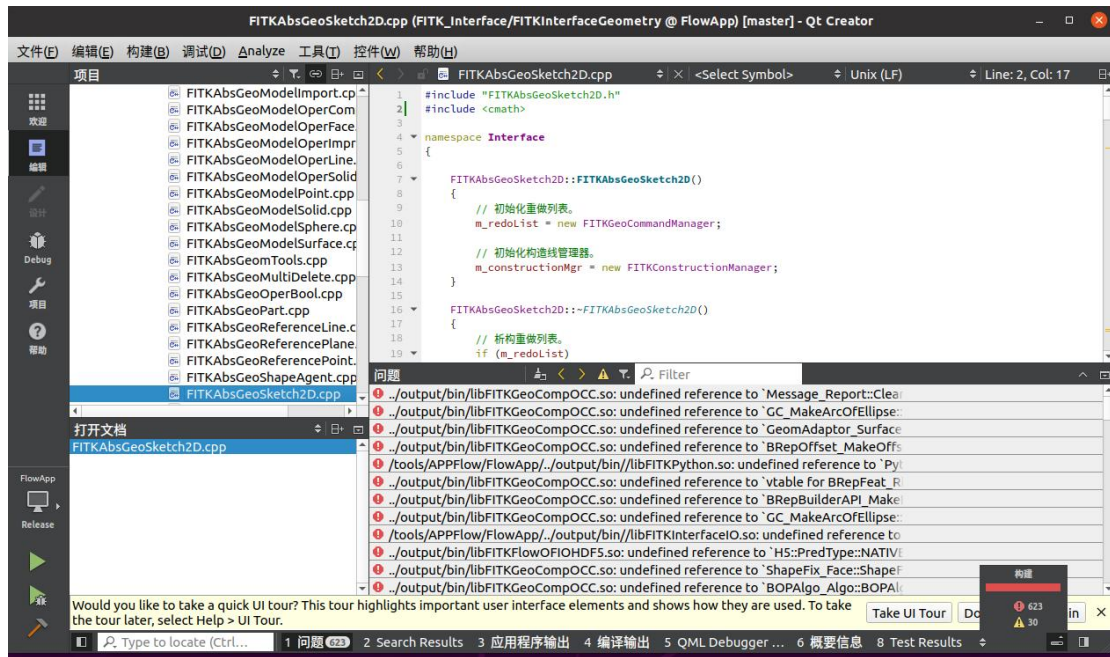
解决方法：在所依赖的项目节点上右键点击，选择构建；构建完成后再点击左下角编译；周而复始，直到这类错误解决完毕。



错误 2.: 编译时未找到第三方库依赖。如下图所示，编译时需要第三方库 qwt 依赖。



和下图所示错误

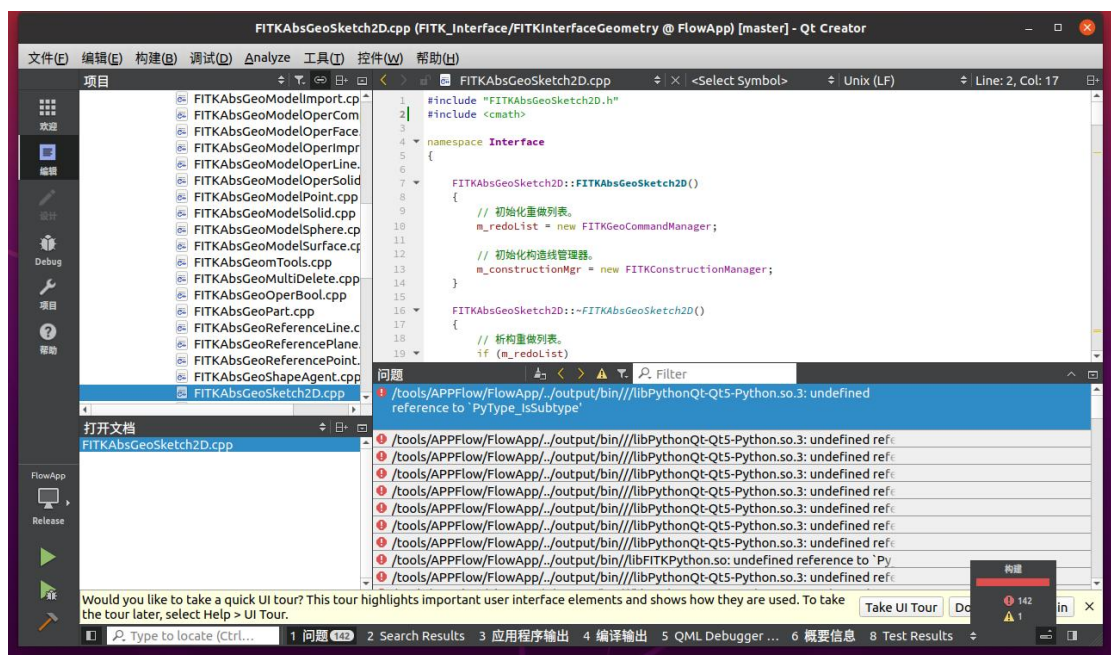


解决方法：

将项目文件/Tools/Linux64/qwt/lib 中的所依赖文件拷贝到项目文件/output/bin 中；再进行编译（其他外部库包括(hdf5 、OCC 、PythonQt、Python37 、qwt 、rapidxml 、SARibbon 、VTK））。

Python37 库比较特殊，需要将其 lib 内的 libpython3.7m.so 、 libpython3.7m.so.1.0 、 libpython3.so 这三个文件放置在 output/bin/目录下。而 Python37 文件夹，则整个放置在、output 文件夹下和其内的 bin 文件夹同级。

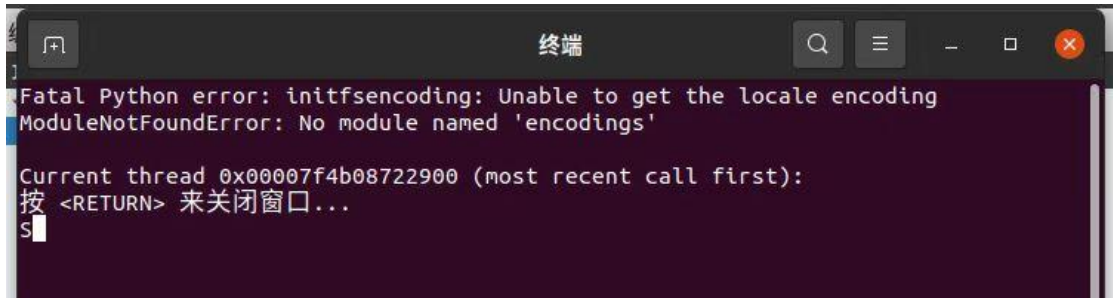
错误 3：





这类报错，可以检查 Python37 库 lib 内的 libpython3.7m.so 、 libpython3.7m.so.1.0 、 libpython3.so 有无放置在 output/bin 文件夹内。

错误 4:

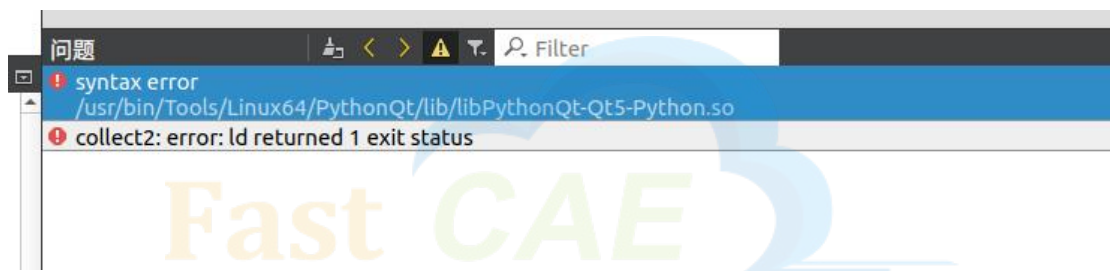


```
Fatal Python error: initfsencoding: Unable to get the locale encoding
ModuleNotFoundError: No module named 'encodings'

Current thread 0x00007f4b08722900 (most recent call first):
按 <RETURN> 来关闭窗口...
S
```

这个错误的解决方法可以考虑 将 “AppFlow/Tools/Linux64/Python37” ， “Python37” 文件夹整个放置在 “APPFlow/output/” 目录下，注意 “Python37” 文件夹应和 “output 文件夹” 中的 “bin 文件夹” 同级。

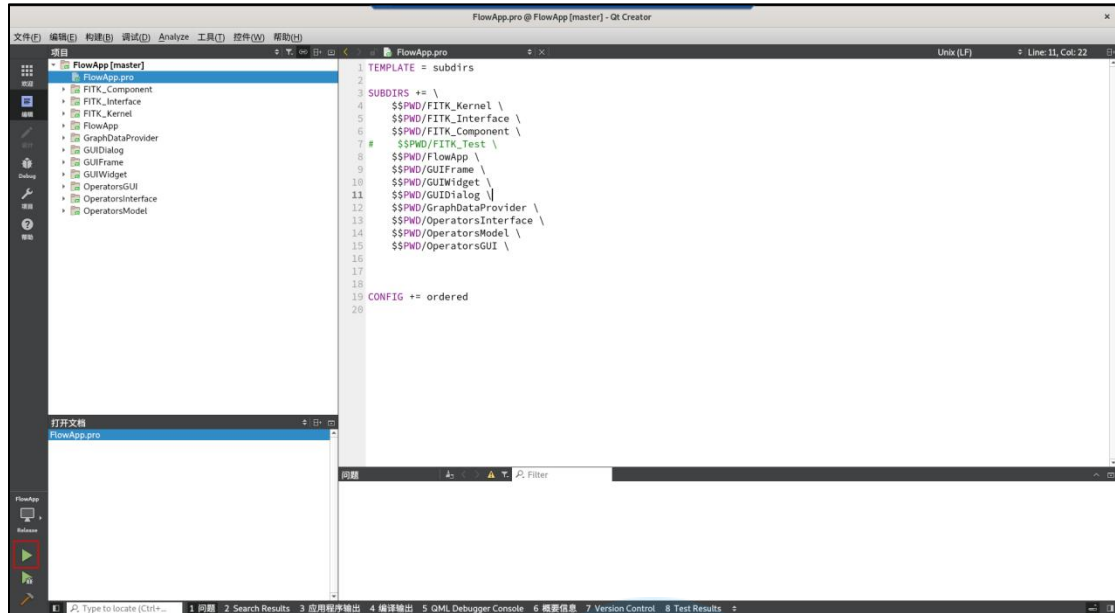
错误 6:



该问题是 qt 将组件识别有误。重新通过 git 拉取项目代码

## 2.8 启动软件

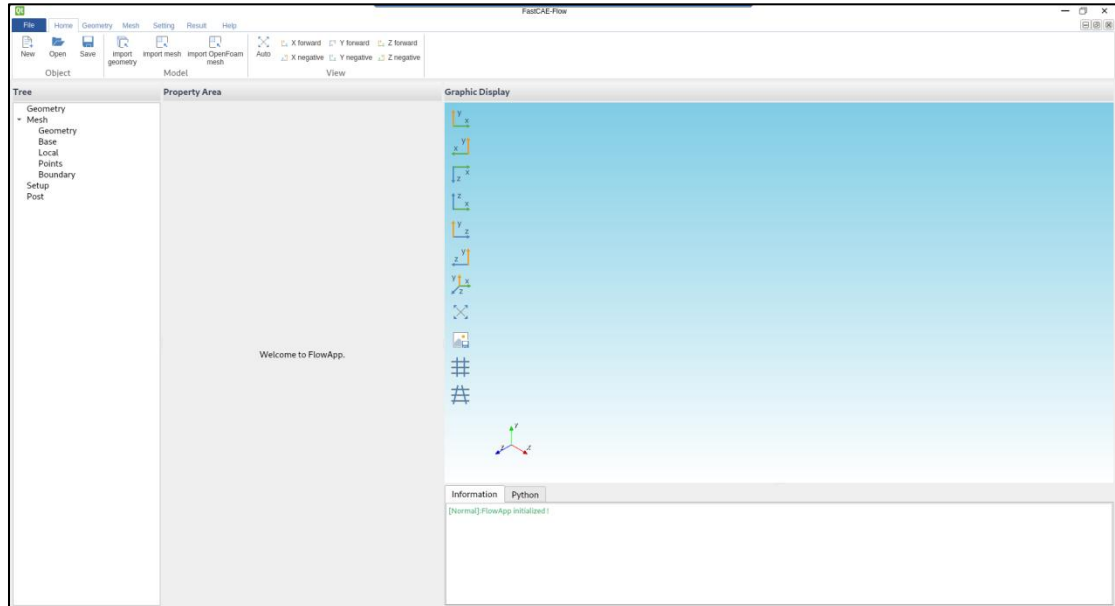
上述编译成功后，点击 Qt 左下角启动按钮，启动程序。



Fast CAE

## 4. 软件启动测试

软件成功启动后的界面如图所示。可根据算例库中提供的算例进行软件使用。



Fast CAE