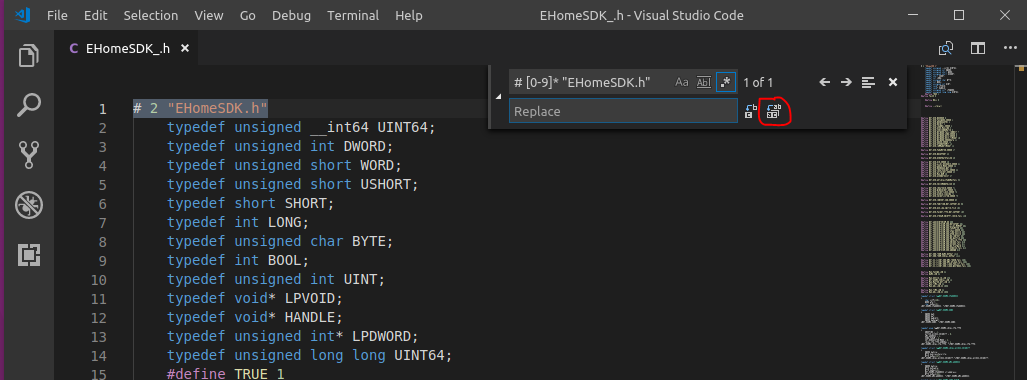
1.要使用doxygen产生无冗余信息的软件模型，首先需要清洗代码，去除冗余信息，具体步骤：

1.1 去掉代码中的注释，注释会导致产生xml模型中包含大量冗余信息，使模型体积增大，增加检索时的内存开销。方法是利用gcc编译器的预处理选项，如下：

gcc -fpreprocessed -dD -E EHomeSDK.h > EHomeSDK\_no\_comment.h

然而，此时产生的文件中会出现大量如 # 2 "EHomeSDK.h" 这样的语句，因此使用vscode编辑器的正则表达式替换为空，如下图：



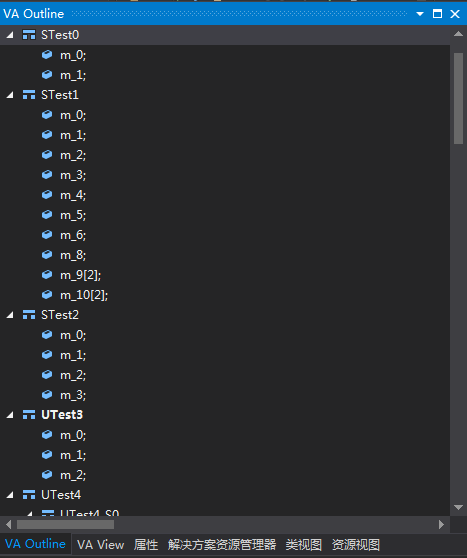
1.2 去掉外部接口的修饰符，如HCNetSDK头文件中为了说明接口为导出符号，需要用NET\_DVR\_API和\_\_stdcall修饰，因此以1.1相同的方式在vscode中批量替换掉修饰符，减少解析模型时的干扰。

1.3注意条件编译代码，若代码中出现条件编译的代码，可根据需要打开条件编译宏。比如windows平台中，PLAYRECT，DWORD，WORD，\_\_int64等为原生类型，其他平台下则没有，因此为了能够让doxygen能够建立这些对象的软件模型，让模型成为闭包，最后将代码中使用到的平台相关的数据类型补全到头文件代码中。

2.因为doxygen工具的问题，其产生的xml静态模型不能确定复合数据类型中的匿名内联复合类型，为了解决此问题，引入outline信息，作为产生复合数据类型软件模型的辅助信息。outline信息的产生步骤如下：

2.1 将头文件中的复合数据类型拷贝出来，也可以去掉其他信息只留下复合数据类型。

2.2打开visual studio并打开Visual AssistX工具的VA Outline视图窗口，如下图：

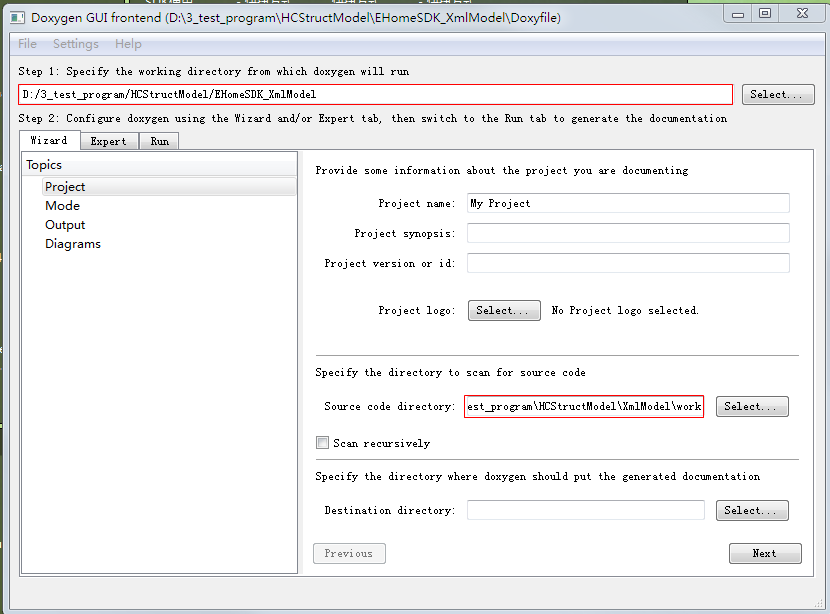


点击右键，选中Copy Outline Content菜单项，即复制Outline成功，然后再粘贴到xxx.outline文件中即可。

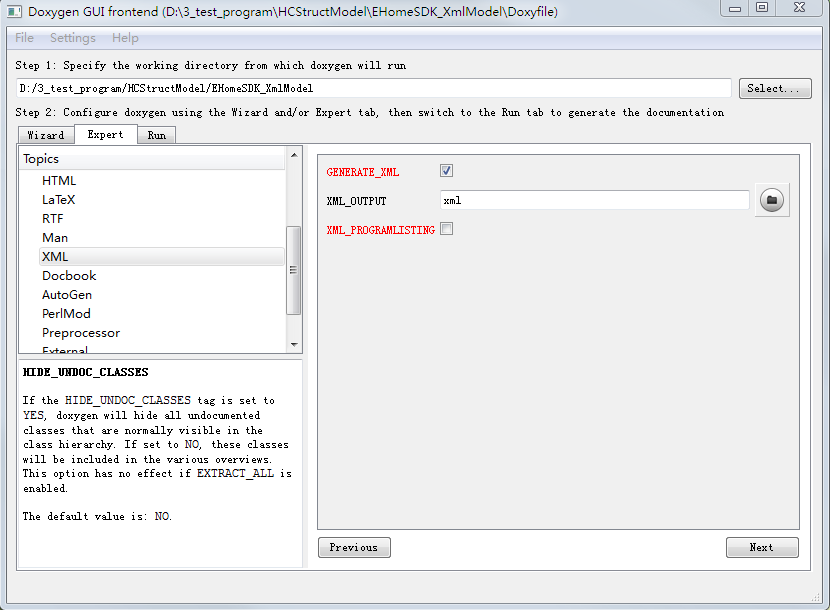
3.到目前为止，我们已经得到了，不含冗余信息的头文件，以及头文件复合数据类型的outline信息，接下来就可以利用doxygen生成静态软件模型了。执行步骤如下：

3.1 下载安装doxygen工具，<http://www.doxygen.nl/download.html>

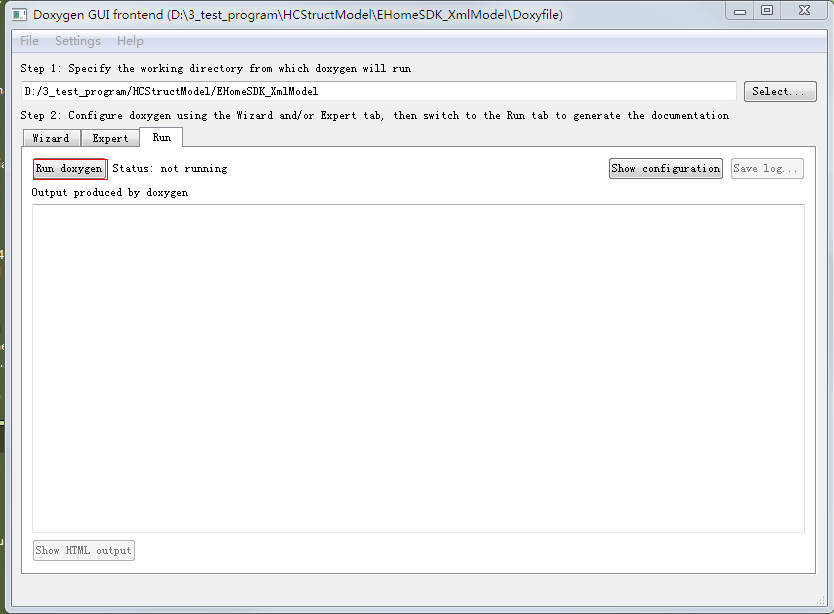
3.2 打开doxygen GUI界面，运行产生静态软件模型，如下：



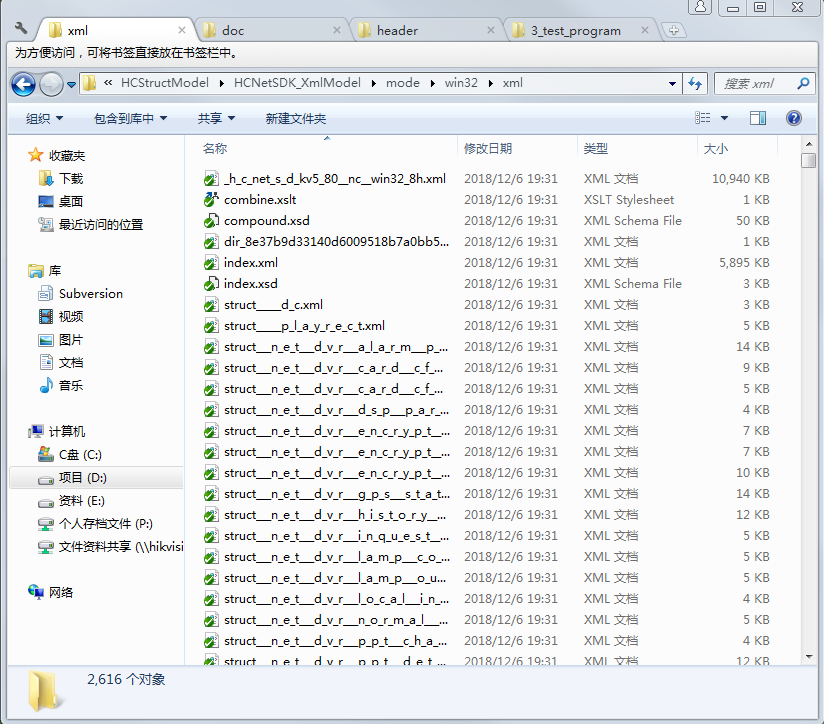
上面红框填写模型存放目录路径，下红框中填写源代码的目录路径，为了标准化请将头文件名设置为work.h。由于我们只需要xml静态模型，因此需要勾选xml生成选项，另外不要勾选最后一项，否则产生的xml文件中含有大量与软件模型无关的冗余信息。



配置完成后，切换到运行table页，点击run按钮产生静态模型，



成功产生软件模型时，文件如下，



4.目前c++建模工具，并没有开发GUI界面，所有操作均在命令行下操作。如，希望获取APINET\_DVR\_ZeroStopPlay的软件模型，步骤如下：

4.1先打开...\xml\work\_8h.xml文件，搜索APINET\_DVR\_ZeroStopPlay，查找结果如下：

|  |
| --- |
| <memberdef kind="function" id="work\_8h\_1a76be0cd270de1f86009bbd016d8cb4da" prot="public" static="no" const="no" explicit="no" inline="no" virt="non-virtual">  <type>BOOL</type>  <definition>BOOL NET\_DVR\_ZeroStopPlay</definition>  <argsstring>(LONG lPlayHandle)</argsstring>  <name>NET\_DVR\_ZeroStopPlay</name>  <param>  <type>LONG</type>  <declname>lPlayHandle</declname>  </param> |

获取其索引ID work\_8h\_1a76be0cd270de1f86009bbd016d8cb4da，然后修改/test/function\_test\_parser.py中test\_find\_fn\_by\_id函数的，如下：

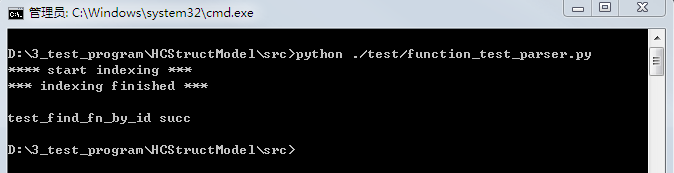
|  |
| --- |
| fn = HFn.parser\_by\_id("work\_8h\_1a76be0cd270de1f86009bbd016d8cb4da",  g\_mode\_dir + "work\_8h.xml") |

4.2配置工作路径，打开/test/test\_config.py文件，修改test\_src\_path值为\HCStructModel\src的绝对路径

4.3保存后，打开terminal，执行如下指令：

|  |
| --- |
| cd ...\\HCStructModel\\src  python ./test/function\_test\_parser.py |

显示如下，则表明成功解析一个函数对象，其他对象解析类似。



通过断点，我们可以查看解析结果，如下图：



BOOL NET\_DVR\_ZeroStopPlay(LONG lPlayHandle)，对比API函数原型，解析结果正确。