

HFD_Mini 力反馈设备在 iMSTK 中的配置说明

一、HFD_Mini 提供的库文件与头文件

由于目前 iMSTK 只支持 x64 的版本，因而 HFD_Mini 亦提供 x64 位平台下 debug 和 release 两种配置的库文件，供 iMSTK 中接口调用，对应的库文件如下所示。

Platform	Configuration			
X64	debug		release	
	CLAF_API64d.lib	CLAF_API64d.dll	CLAF_API64.lib	CLAF_API64.dll

提供的头文件：CLAF_OPEN.h 、 clafDefines.h 、 clafVector.h、 clafVector.inl

二、iMSTK 中新建的文件

为了在 iMSTK 库中适配 HFD_Mini 力反馈设备，构建了两个类：CLAFHapticDeviceManager 与 CLAFHapticDeviceClient。分别位于如下文件中：

imstkCLAFHapticDeviceManager.h

imstkCLAFHapticDeviceManager.cpp

imstkCLAFHapticDeviceClient.h

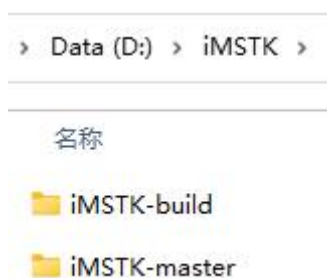
imstkCLAFHapticDeviceClient.cpp

CLAFHapticDeviceManager 类是对 HFD_Mini 设备进行统一管理的类，其在应用程序中被直接调用，包括 imstkCLAFHapticDeviceManager.h 与 imstkCLAFHapticDeviceManager.cpp 两个文件。

CLAFHapticDeviceClient 类是对 HFD_Mini 设备的功能进行直接的调用，其在内部会调用 CLAF_API64 库中的 API 接口，包括 imstkCLAFHapticDeviceClient.h 与 imstkCLAFHapticDeviceClient.cpp 两个文件。

三、具体的配置步骤

iMSTK 库的源文件位于 iMSTK-master 文件夹中，通过 Cmake 构建生成后的文件位于 iMSTK-build 文件夹中。如下图所示。



Step1： 将 imstkCLAFHapticDeviceManager.h、imstkCLAFHapticDeviceManager.cpp、imstkCLAFHapticDeviceClient.h、imstkCLAFHapticDeviceClient.cpp 四个文件拷贝到 iMSTK-master>Source>Devices 文件夹下，如下图所示。

> Data (D:) > iMSTK > iMSTK-master > Source > Devices

名称	修改日期	类型	大小
CMakeLists.txt	2023/2/8 4:02	文本文档	2 KB
imstkCLAFHapticDeviceClient.cpp	2023/2/25 19:13	C++ Source	7 KB
imstkCLAFHapticDeviceClient.h	2023/2/24 16:56	C/C++ Header	3 KB
imstkCLAFHapticDeviceManager.cpp	2023/2/24 16:56	C++ Source	2 KB
imstkCLAFHapticDeviceManager.h	2023/2/24 16:56	C/C++ Header	2 KB

Step2： 在 iMSTK-build>install>include 文件夹下新建文件夹 CLAF，将 CLAF_OPEN.h、clafDefines.h、clafVector.h、clafVector.inl 四个文件放置在新建的 CLAF 文件夹下，如下图所示。

> Data (D:) > iMSTK > iMSTK-build > install > include > CLAF

名称	修改日期	类型	大小
CLAF_OPEN.h	2021/9/14 16:21	C/C++ Header	17 KB
clafDefines.h	2022/4/18 14:37	C/C++ Header	14 KB
clafVector.h	2021/8/9 17:43	C/C++ Header	5 KB
clafVector.inl	2021/8/9 17:44	C/C++ Inline File	8 KB

Step3： 将 CLAF_API64.lib 与 CLAF_API64d.lib 两个库文件拷贝到 iMSTK-build>install>lib 文件夹下，如下图所示。

> Data (D:) > iMSTK > iMSTK-build > install > lib

名称	修改日期	类型	大小
CLAF_API64.lib	2022/12/16 11:43	Object File Library	38 KB
CLAF_API64d.lib	2022/6/15 10:06	Object File Library	39 KB


Step4： 将 CLAF_API64.dll 与 CLAF_API64d.dll 两个动态链接库文件拷贝到 iMSTK-build>install>bin 文件夹下，如下图所示。

» Data (D:) » iMSTK » iMSTK-build » install » bin

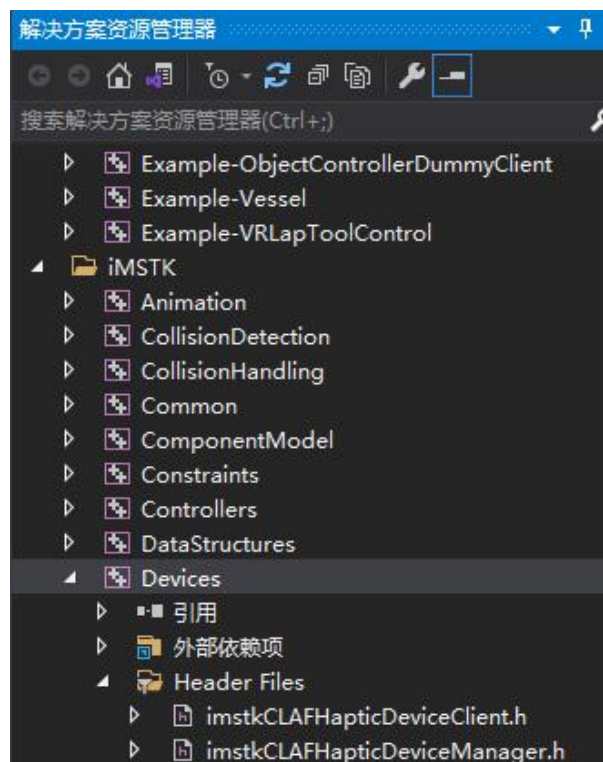
名称	修改日期	类型	大小
 CLAF_API64.dll	2022/12/16 11:43	应用程序扩展	188 KB
 CLAF_API64d.dll	2022/12/16 11:46	应用程序扩展	344 KB

Step5: 用 VisualStudio2019 打开 iMSTK-build>Innerbuild 文件夹下的 iMSTK.sln 解决方案文件，如下图所示。

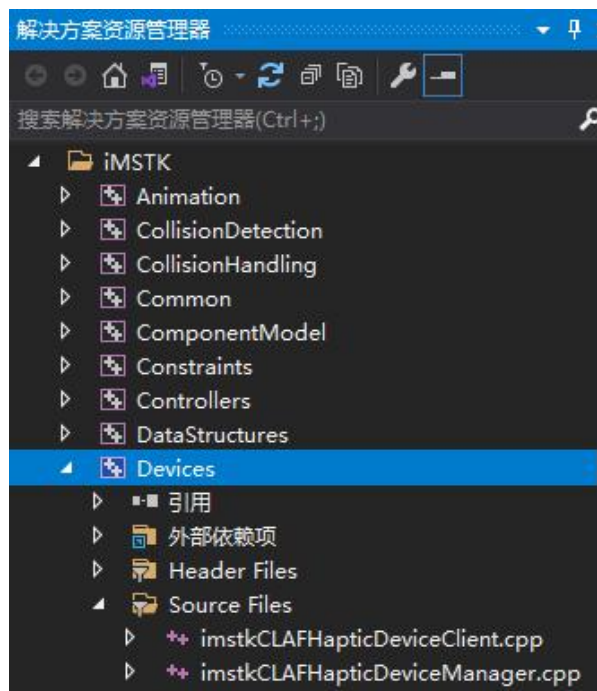
» Data (D:) » iMSTK » iMSTK-build » Innerbuild

名称	修改日期	类型	大小
 CopyDataFiles.vcxproj	2023/2/14 12:03	VC++ Project	40 KB
 CopyDataFiles.vcxproj.filters	2023/2/14 23:52	VC++ Project Filters F...	1 KB
 CTestTestfile.cmake	2023/2/14 23:52	CMAKE 文件	6 KB
 DartConfiguration.tcl	2023/2/25 12:59	TCL 文件	3 KB
 Experimental.vcxproj	2023/2/25 12:03	VC++ Project	40 KB
 Experimental.vcxproj.filters	2023/2/14 23:52	VC++ Project Filters F...	1 KB
 iMSTK.sln	2023/2/14 23:52	Visual Studio Solution	273 KB

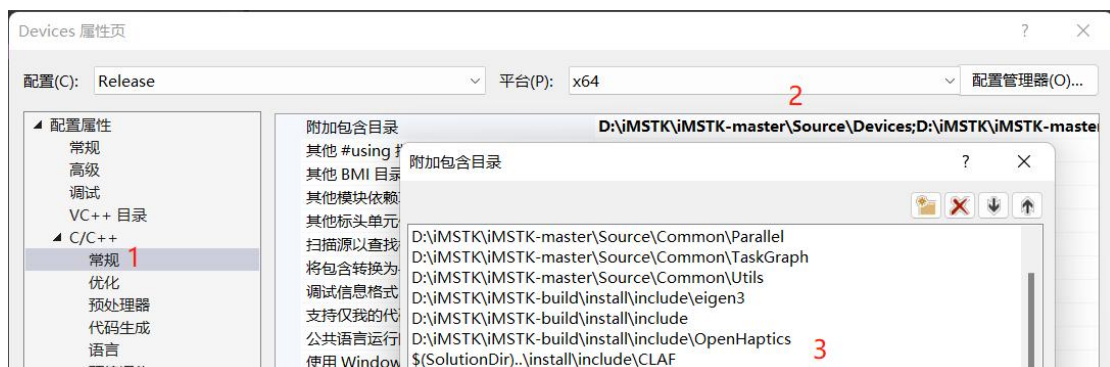
Step6: 将 imstkCLAFHapticDeviceManager.h 与 imstkCLAFHapticDeviceClient.h 文件添加到 iMSTK 文件夹下 Devices 项目的 Header Files 中，如下图所示。



Step7: 将 imstkCLAFHapticDeviceManager.cpp 与 imstkCLAFHapticDeviceClient.cpp 文件添加到 iMSTK 文件夹下 Devices 项目的 Source Files 中，如下图所示。



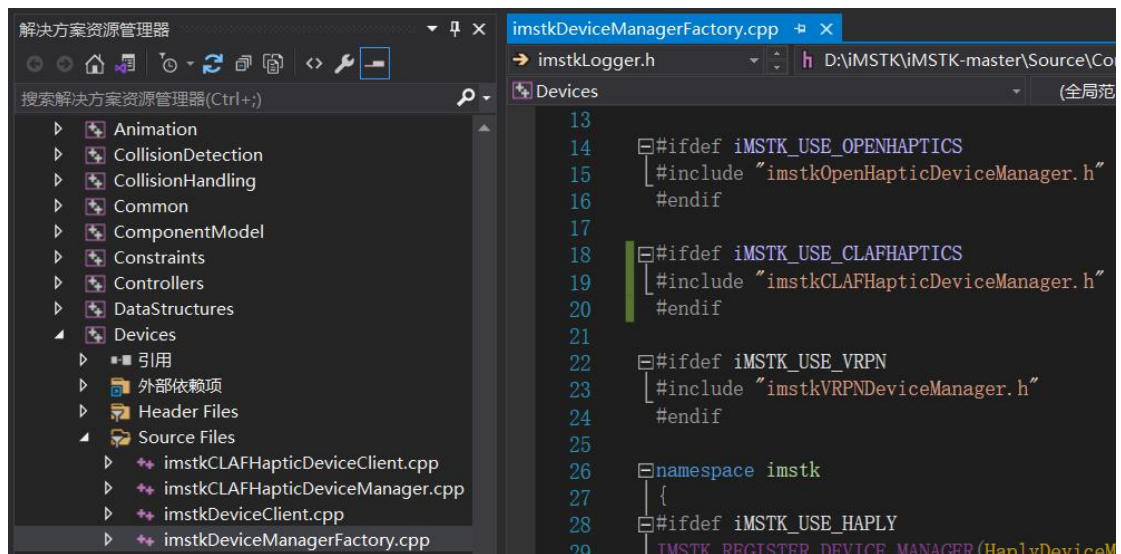
Step8: 打开 Devices 属性对话框，在配置属性→C/C++→常规→附加包含目录中，添加 CLAF 文件夹，可以是绝对路径，也可以是相对路径，如下图所示。



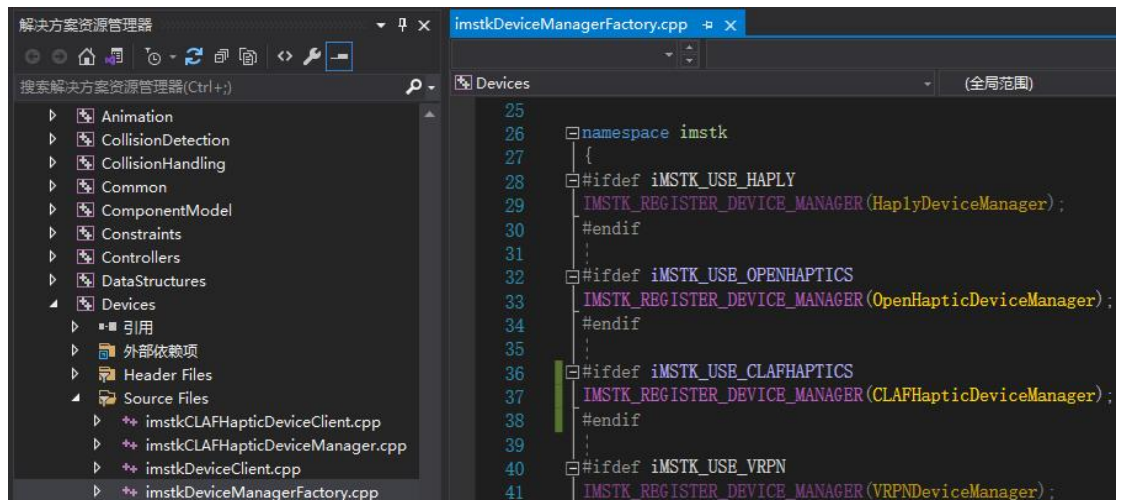
Step9: 在打开的 Devices 属性对话框中，在配置属性→C/C++→预处理器→预处理器定义中，添加定义：iMSTK_USE_CLAFHAPTICS，如下图所示。



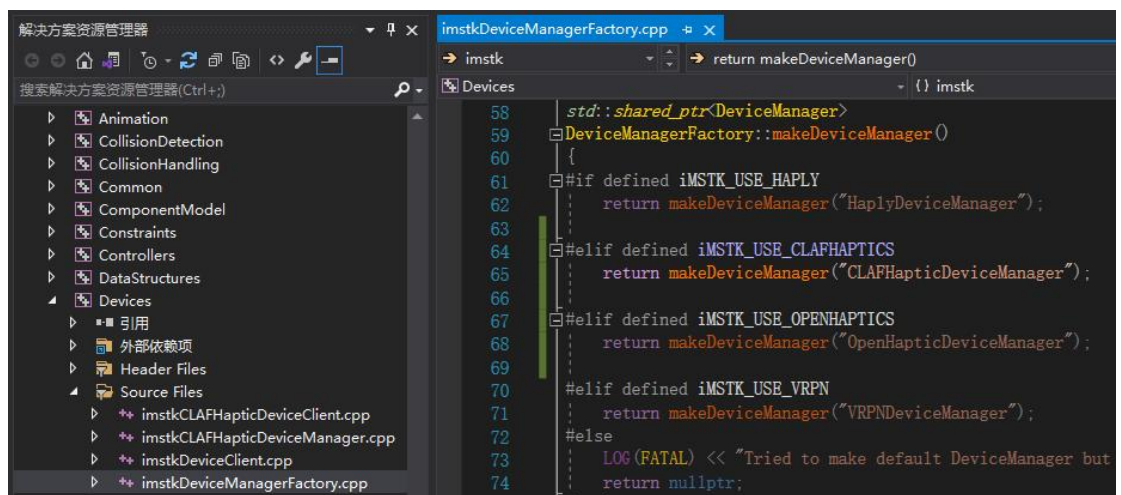
Step10: 打开 Devices 项目下的 imstkDeviceManagerFactory.cpp 文件，在文件中添加条件编译代码，如下图所示。



Step11: 打开 Devices 项目下的 imstkDeviceManagerFactory.cpp 文件，在文件中添加如下条件编码代码，如下图所示。

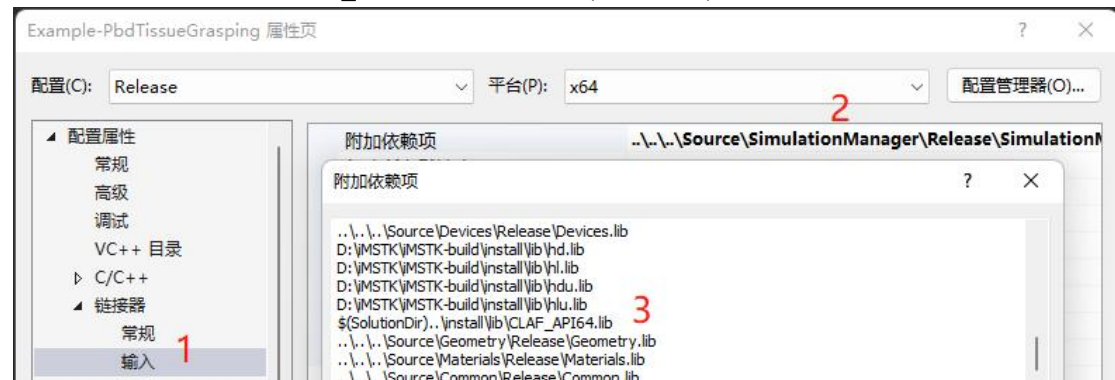


Step12: 打开 Devices 项目下的 imstkDeviceManagerFactory.cpp 文件，在 makeDeviceManager 函数中添加如下代码，并且要添加在 OPENHAPTICS 相关代码之前，如下图所示。



Step13: 到此，Devices 项目中的操作完成，重新生成 debug 或者 release 配置下的 Devices 项目，对应地，会生成 Devicesd.lib 和 Device.lib 静态链接库。

Step14: 在要测试有 Example 下的项目中，打开属性对话框，在配置属性→链接器→输入→附加依赖项中，输入 CLAF_API64.lib 库的路径(包含文件)，可以是相对路径 或 绝对路径。



Step15: 重新生成该 Example 可执行项目，然后即可正常打开运行，测试 HFD_Mini 力反馈设备。