

Configuration Guide for HFD_6 in iMSTK

1. Library and Header Files Porvided by HFD_6

Since iMSTK currently supports only x64 platforms, HFD_6 provides x64 library files for both **Debug** and **Release** configurations, as shown below:

Platform	Configuration			
	debug		release	
X64	HFD_API64d.lib	HFD_API64d.dll	HFD_API64.lib	HFD_API64.dll

Header Files: HFD_OPEN.h 、 hfdDefines.h 、 hfdVector.h、 hfdVector.inl

2. New Files in iMSTK

To integrate the HFD-6 device into iMSTK , two classes have been developed: HFDHapticDeviceManager and HFDHapticDeviceClient:

- imstkHFDHapticDeviceManager.h
- imstkHFDHapticDeviceManager.cpp
- imstkHFDHapticDeviceClient.h
- imstkHFDHapticDeviceClient.cpp

HFDHapticDeviceManager manages the HFD-6 device at the application level,including: **imstkHFDHapticDeviceManager.h** and **imstkHFDHapticDeviceManager.cpp**.

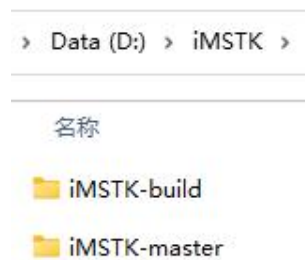
HFDHapticDeviceClient directly interfaces with HFD-6 via the **HFD_API64** library, including: **imstkHFDHapticDeviceClient.h** and **imstkHFDHapticDeviceClient.cpp**.

3. Configuration Steps

Project Structure:

- iMSTK source files: Located in the **iMSTK-master** folder.
- CMake-generated files: Located in the **iMSTK-build** folder.

As shown in the figure below.



Step1: Copy Class Files

Copy the following files to **iMSTK-master>Source>Devices**:

imstkHFDHapticDeviceManager.h

imstkHFDHapticDeviceManager.cpp

imstkHFDHapticDeviceClient.h

imstkHFDHapticDeviceClient.cpp

As shown below.

> 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: Configure Header Files

Create a folder named **HFD** under **iMSTK-build>install>include**.

Place the following header files into this folder:

- **HFD_OPEN.h**
- **hfdDefines.h**
- **hfdVector.h**
- **hfdVector.inl**

As shown below.

> 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: Configure Library Files



Copy the following library files to **iMSTK-build>install>lib**:

- **HFD_API64.lib**

● HFD_API64d.lib

As shown below.

› 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: Configure DLL Files

Copy the following DLL files to **iMSTK-build>install>bin**:

● HFD_API64.dll

● HFD_API64d.dll

As shown below.

› 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: Open iMSTK Solution in Visual Studio 2019

1. Navigate to **iMSTK-build>Innerbuild**.

2. Open the **iMSTK.sln** solution file.

As shown below.

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

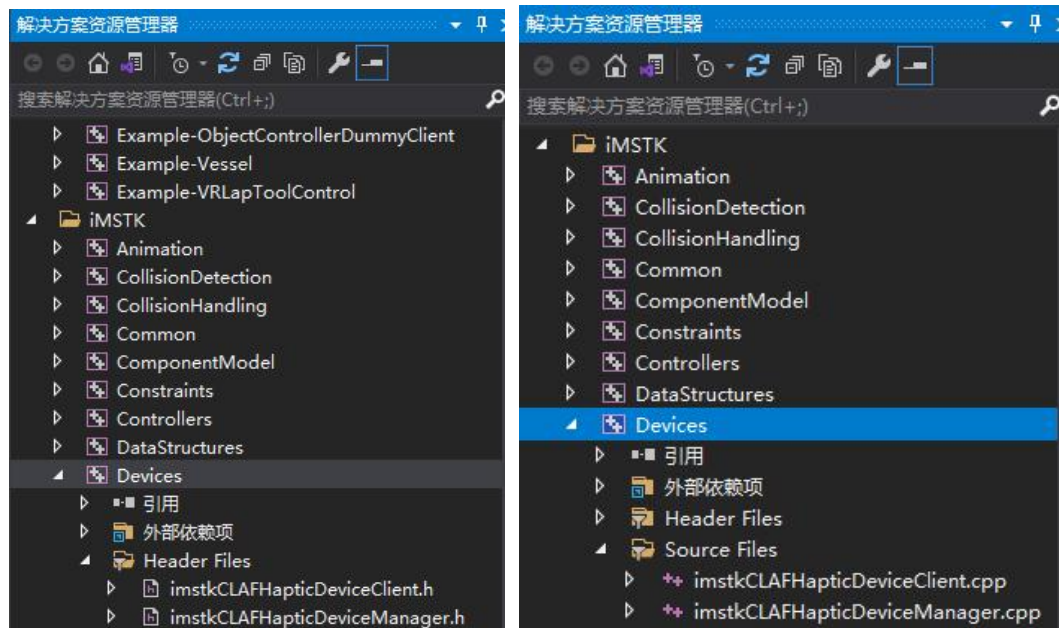
名称	修改日期	类型	大小
 CopyDataFiles.vcxproj	2023/2/14 23:52	VC++ Project	40 KB
 CopyDataFiles.vcxproj.filters	2023/2/14 23:52	VC++ Project Filters F...	1 KB
 CTestTestfile.cmake	2023/2/14 23:52	CMAKE 文件	6 KB
 DartConfiguration.tcl	2023/2/25 12:59	TCL 文件	3 KB
 Experimental.vcxproj	2023/2/25 12:03	VC++ Project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

1. **Header Files**: Add **imstkHFDHapticDeviceManager.h** and **imstkHFDHapticDeviceClient.h** to the **Header Files** section under the **Devices** Project.

2. **Source Files**: Add **imstkHFDHapticDeviceManager.cpp** and **imstkHFDHapticDeviceClient.cpp** to **Source Files** section.

As shown below.



Step8: Configure Project Properties

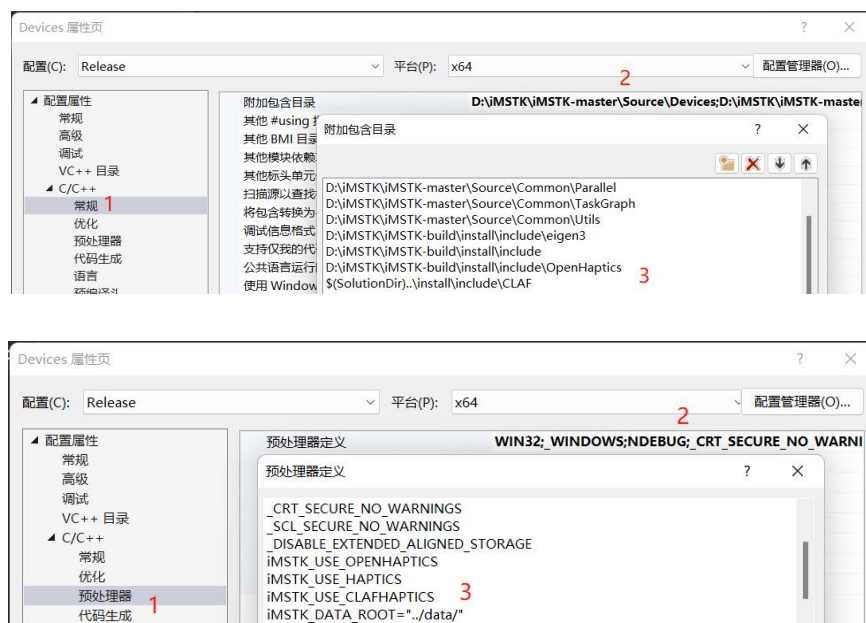
1. Additional Include Directories:

- Navigate to **Properties > C/C++ > General > Additional Include Directories**.
- Add the path to the **HFD** folder (absolute or relative).

2. Preprocessor Definitions:

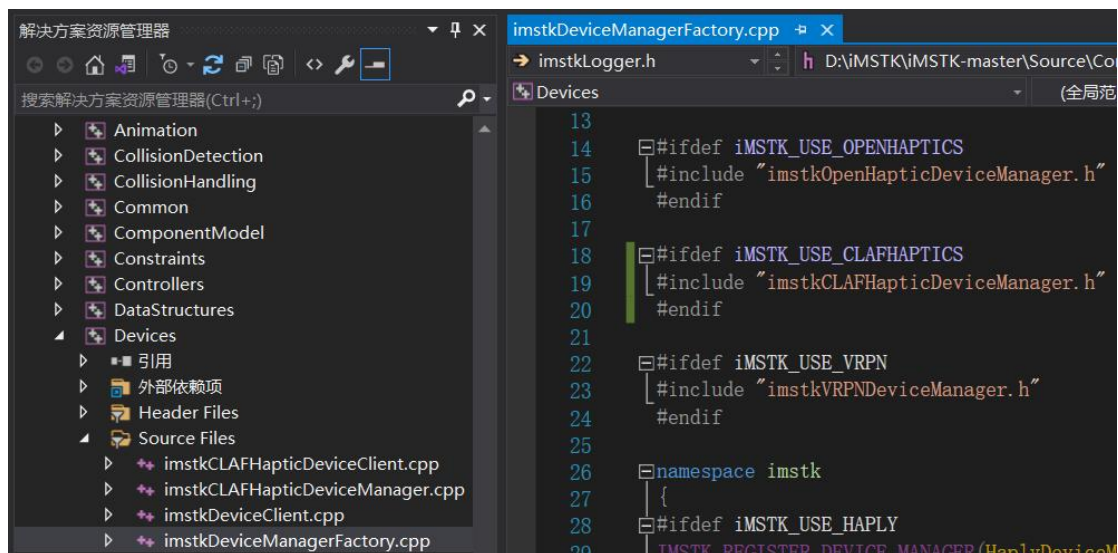
- Navigate to **Properties > C/C++ > Preprocessor > Preprocessor Definitions**.
- Add **iMSTK_USE_HFDHAPTICS**.

As shown below.

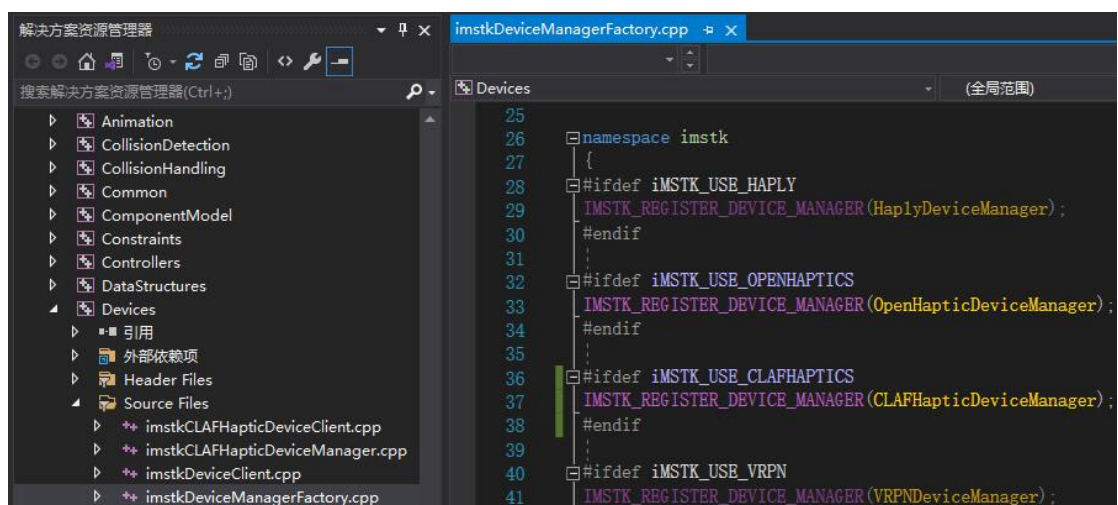


Step9: Modify Source Code

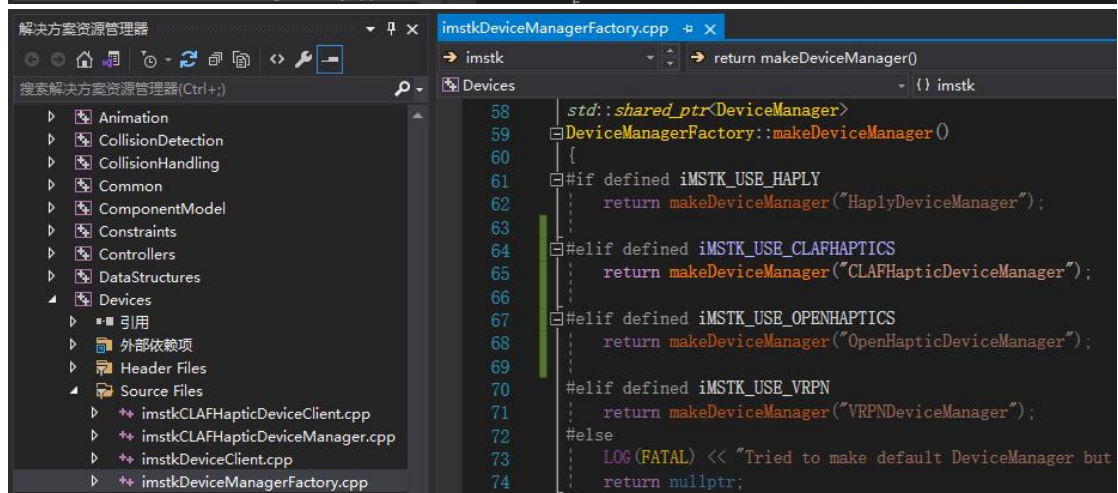
1. Open **imstkDeviceManagerFactory.cpp** under the **Devices** project.
2. Add conditional compilation code before the OPENHAPTICS-related code in the **makeDeviceManager** function.



```
13
14 #ifndef iMSTK_USE_OPENHAPTICS
15 #include "imstkOpenHapticDeviceManager.h"
16 #endif
17
18 #ifndef iMSTK_USE_CLAFHAPTICS
19 #include "imstkCLAFHapticDeviceManager.h"
20 #endif
21
22 #ifndef iMSTK_USE_VRPN
23 #include "imstkVRPNDeviceManager.h"
24 #endif
25
26 namespace imstk
27 {
28 #ifndef iMSTK_USE_HAPLY
29 IMSTK_REGISTER_DEVICE_MANAGER(HaplyDeviceM
```



```
25
26 namespace imstk
27 {
28 #ifndef iMSTK_USE_HAPLY
29 IMSTK_REGISTER_DEVICE_MANAGER(HaplyDeviceManager);
30 #endif
31
32 #ifndef iMSTK_USE_OPENHAPTICS
33 IMSTK_REGISTER_DEVICE_MANAGER(OpenHapticDeviceManager);
34 #endif
35
36 #ifndef iMSTK_USE_CLAFHAPTICS
37 IMSTK_REGISTER_DEVICE_MANAGER(CLAFHapticDeviceManager);
38 #endif
39
40 #ifndef iMSTK_USE_VRPN
41 IMSTK_REGISTER_DEVICE_MANAGER(VRPNDeviceManager);
```



```
58 std::shared_ptr<DeviceManager>
59 DeviceManagerFactory::makeDeviceManager()
60 {
61 #if defined iMSTK_USE_HAPLY
62 return makeDeviceManager("HaplyDeviceManager");
63 #elif defined iMSTK_USE_CLAFHAPTICS
64 return makeDeviceManager("CLAFHapticDeviceManager");
65 #elif defined iMSTK_USE_OPENHAPTICS
66 return makeDeviceManager("OpenHapticDeviceManager");
67 #elif defined iMSTK_USE_VRPN
68 return makeDeviceManager("VRPNDeviceManager");
69 #else
70 LOG(FATAL) << "Tried to make default DeviceManager but
71 return nullptr;
```


Step10: Build and Test

1. Rebuild the **Devices** project in Debug/Release mode to generate **Devicesd.lib** or **Devices.lib**.
2. For Example Projects:
 - In the project properties, navigate to **Linker > Input > Additional Dependencies**.
 - Add the path to **HFD_API64.lib**.
3. Rebuild the example project and run it to test the HFD-6 device.

