

Cocos2Dx Integration Guide

Universal Haptic Layer

Version 1.0 | Revision 3

Downloading the UHL SDK Package

Get the UHL package from the [SDK download page](#) and unzip it to a location of your choice.

UHL Package

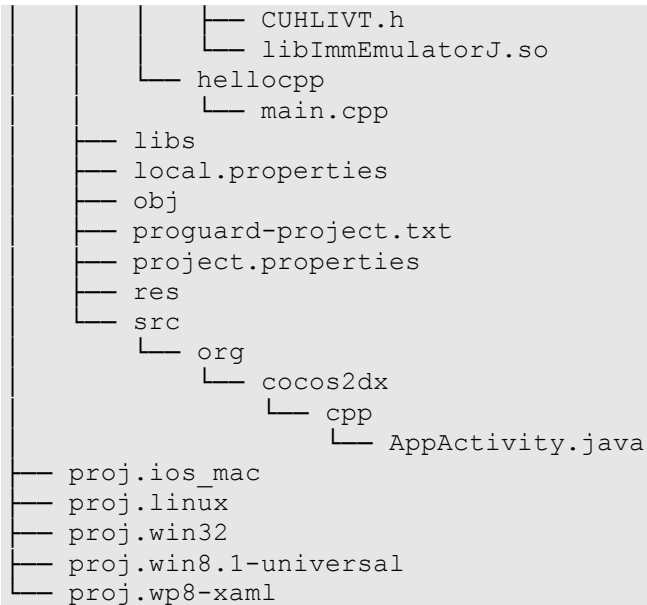
The UHL package is composed of the following components:

- **UHL.jar**: Java library containing the UHL implementation. Java applications that use UHL must reference this library. For Cocos2dx, this jar is not required.
- **UHLdoc.zip**: Documentation in javadoc format.
- **libImmEmulatorJ.so**: Native library containing the TouchSense 3000 Emulator implementation. Applications that use UHL must link with this library. Use the library that coincides with the architecture you are targeting.
- **CUHL Header files .h**: These will need to be included in order for your app to call UHL's various APIs.

Installing and Using UHL Package in Cocos2dx

1. Copy the CUHL header files and libImmEmulatorJ.so into your Cocos2dx Game. We created an additional directory under proj.android/jni named CUHL. Below is a generic Cocos2dx project structure with UHL headers and library added.

```
.
├── CMakeLists.txt
├── Classes
├── Resources
├── proj.android
│   ├── AndroidManifest.xml
│   ├── ant.properties
│   ├── assets
│   ├── bin
│   ├── build-cfg.json
│   ├── build.xml
│   ├── build_native.py
│   └── jni
│       ├── Android.mk
│       ├── Application.mk
│       └── CUHL
│           ├── CUHL.h
│           └── CUHLAPI.h
```



2. Modify your Android.mk to include the prebuilt shared library.

```

LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE := CUHL
LOCAL_SRC_FILES := CUHL/libImmEmulatorJ.so
include $(PREBUILT_SHARED_LIBRARY)
  
```

And within the module that contains your Cocos2dx source code include the CUHL module you just created and the path to the CUHL header files.

```

LOCAL_SHARED_LIBRARIES := CUHL
LOCAL_C_INCLUDES += $(LOCAL_PATH)/CUHL
  
```

3. In order for UHL to function, the Android Vibrate and Internet permission are required. Make sure to add these permissions the **AndroidManifest.xml** found under the *proj.android/* directory.
4. Inside the default **AppActivity** class, or your own custom Android Activity, load the shared library by adding this line in the `onCreate()` method of your Activity.

```

try {
    System.loadLibrary("ImmEmulatorJ");
} catch (UnsatisfiedLinkError e) {
    // error-statement here
}
  
```

5. Be sure to add the following header files, to the `.cpp` file where you plan on using UHL.

```

#include <CUHL.h>
#include <jni/JniHelper.h>
#include <jni.h>
  
```

6. Now, you can initialize the API and open a device. In order to initialize UHL, you will need a reference to the JavaVM as well as an Android Application Context. The following snippet uses JNI to obtain a reference to the current application context object.

```
// UHL requires a reference to the JVM
JavaVM *jvm = NULL;

jvm = JNIHelper::getJavaVM();
if (jvm == NULL) {
    return false;
}

// UHL also requires an Android Context
MethodInfo getContext;

if (!JNIHelper::getStaticMethodInfo(getContext,
    "org/cocos2dx/lib/Cocos2dxActivity",
    "getContext",
    "()Landroid/content/Context;")) {
    return false;
}

 jobject context = getContext.env->CallStaticObjectMethod(getContext.classID,
    getContext.methodID);
```

7. After we have obtained the context, you can use the context object to initialize the UHL API and open a device. An Immersion provided username and password will be required to initialize UHL.

```
// Initialize the API
VibeStatus status = VIBE_E_FAIL;
char *username = ""; /*Immersion provided username*/
char *password = ""; /*Immersion provided password*/
char *dns = ""; /*Immersion provided DNS url. NULL otherwise*/

status = ImmVibeInitialize2(0, jvm, context, username, password, dns);

if (VIBE_FAILED(status)) {
    __android_log_print(ANDROID_LOG_DEBUG, "ImmVibe", "failed to init api");
    return false;
}

// Open Device
VibeInt32 deviceHandle;

status = ImmVibeOpenDevice(0, &deviceHandle);
if (VIBE_FAILED(status)) {
    __android_log_print(ANDROID_LOG_DEBUG, "ImmVibe", "failed to open device");
    return false;
}
```

8. After initializing and opening device, you are then able to play haptics.

```
// Play haptics
VibeInt32 effectHandle;

status = ImmVibePlayUHLEffect(deviceHandle, TRIPLE_STRONG_CLICK_100,
    &effectHandle);
if (VIBE_FAILED(status)) {
    __android_log_print(ANDROID_LOG_DEBUG, "ImmVibe", "failed to play effect");
    return false;
}
```

LEGAL DISCLAIMERS

General Disclaimer.

Although Immersion has attempted to provide accurate information in this document, Immersion assumes no responsibility for the accuracy of the information. Immersion may change the products or services (or information relating thereto) mentioned at any time without notice. Mention of non-Immersion products or services is for information purposes only and constitutes neither an endorsement nor a recommendation.

EXCEPT WHERE EXPRESSLY PROVIDED OTHERWISE, THIS DOCUMENT, INCLUDING ALL CONTENT, MATERIALS, INFORMATION, SOFTWARE, PRODUCTS AND SERVICES DESCRIBED THEREIN, ARE PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS. IMMERSION EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT AND WARRANTIES ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE.

IMMERSION SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL OR INCIDENTAL DAMAGES, INCLUDING WITHOUT LIMITATION, LOST PROFITS OR REVENUES, COSTS OF REPLACEMENT GOODS, LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT OR ANY IMMERSION PRODUCT, DAMAGES RESULTING FROM USE OF OR RELIANCE ON CONTENT, EVEN IF IMMERSION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. YOUR SOLE REMEDY FOR DISSATISFACTION WITH THIS DOCUMENT AND/OR THE INFORMATION CONTAINED HEREIN IS TO CEASE ALL OF YOUR USE OF THIS DOCUMENT AND SUCH INFORMATION. To the extent required by law in other jurisdictions, certain of the above liability limitations or warranty exclusions may not apply to you. You agree that the warranties and the liability of Immersion would in such case be limited to the greatest extent permitted by law. The information in this document may be changed at any time.

Intellectual Property.

The trademarks, logos and service marks ("Marks") displayed in this document are the property of Immersion or other third parties. You are not permitted to use these Marks without the prior written consent of Immersion or such appropriate third party.

All information in this document is (and shall continue to be) owned exclusively by Immersion or other third parties owners, and is protected under applicable copyrights, patents, trademarks, trade dress, and/or other proprietary rights, and the copying, redistribution, use or publication by you of any such information or any part of this document is prohibited. Under no circumstances will you acquire any ownership rights or other interest in any information contained herein by or through your use of this document.

©2016 Immersion Corporation. All rights reserved. Immersion, the Immersion logo, and TouchSense are trademarks of Immersion Corporation in the United States and other countries. All other trademarks are the property of their respective owners.



immersion.com | 408.467.1900
50 Rio Robles | San Jose, California 95134