



Using the Logitech Gaming LED SDK with Epic Games' UDK and DLLBind

© 2012 Logitech. Confidential

The Logitech Gaming LED SDK, including all accompanying documentation, is protected by intellectual property laws. All use of the Logitech Gaming LED SDK is subject to the License Agreement found in the "Logitech Gaming LED SDK License Agreement" file and at the end of this document. If you do not agree to the terms and conditions of the License Agreement, you must immediately return any documentation, the accompanying software and all other material provided to you by Logitech. All rights not expressly granted by Logitech are reserved.

Contents

Overview	3
Making the LED SDK work in your UDK game	3
Steps	3
Calling Logitech SDK's functions from within the game.....	4

Overview

The Logitech Gaming LED SDK enables applications such as games to control the backlight LEDs on supported Logitech gaming mice and keyboards.

It can be easily integrated in a UDK game by using DLLBind.

Please refer to the Logitech SDK's Doc\LogitechGamingLEDSDK.pdf for details on the SDK's functionality.

Making the LED SDK work in your UDK game

The following steps show how to make the Logitech SDK work with "UDKGame" that comes as part of the UDK download. Please adapt the steps to your game for things to work.

Steps

1. Download UDK (<http://www.unrealengine.com/udk/>).
2. Create the following 2 Unreal Script files, and copy them to UDK's Development\Src\MyMod\Classes folder:

TestDLLGameInfo.uc

```
class TestDLLGameInfo extends GameInfo;

defaultproperties
{
    PlayerControllerClass=class'TestDLLPlayerController'
}
```

TestDLLPlayerController.uc

```
class TestDLLPlayerController extends PlayerController
    DLLBind(LogitechLed);

dllimport final function bool LogiLedInit();
dllimport final function bool LogiLedSaveCurrentLighting(int deviceType);
dllimport final function bool LogiLedSetLighting(int deviceType, int
redPercentage, int greenPercentage, int bluePercentage);
dllimport final function bool LogiLedRestoreLighting(int deviceType);
dllimport final function LogiLedShutdown();

exec function LogiInit()
{
    local bool ret;
    ret = LogiLedInit();

    say("LogiLedInit return is: " $ret);
}

exec function LogiSaveCurrentLighting(int deviceType)
{
    local bool ret;
```

```
        ret = LogiLedSaveCurrentLighting(deviceType);

        say("LogiLedSaveCurrentLighting return is: " $ret);
    }

exec function LogiSetLighting(int deviceType, int redPercentage, int
greenPercentage, int bluePercentage)
{
    local bool ret;
    ret = LogiLedSetLighting(deviceType, redPercentage, greenPercentage,
bluePercentage);

    say("LogiLedSetLighting return is: " $ret);
}

exec function LogiRestoreLighting(int deviceType)
{
    local bool ret;
    ret = LogiLedRestoreLighting(deviceType);

    say("LogiLedRestoreLighting return is: " $ret);
}

exec function LogiShutdown()
{
    LogiLedShutdown();

    say("LogiLedShutdown done");
}
```

3. Copy Logitech SDK's Lib\x86\ LogitechLed.dll to UDK's Binaries\Win32\UserCode
4. Copy Logitech SDK's Lib\x64\ LogitechLed.dll to UDK's Binaries\Win64\UserCode
5. Open UDK's UDKGame\Config\ DefaultEngineUDK.ini file for editing
 - a. Search for: `ModEditPackages=MyMod`
 - b. Remove the `;` at the beginning of the line
6. Launch UDK's Binaries/UnrealFrontend.exe
 - a. Do: Script->Full recompile

Calling Logitech SDK's functions from within the game

Launch game the following way:

- Binaries\Win32\UDK.exe dm-deck?game=MyMod.TestDLLGameInfo

Once the game is running, open the console (hit the ~ key), and type: `LogiInit`, and then hit <enter> key. You should see the " LogiLedInit return is: TRUE" message.

Then use the other commands as defined in the Unreal Script file:

- `LogiSetLighting`
- `LogiSaveCurrentLighting`
- `LogiRestoreLighting`
- `LogiShutdown`

Using the LED SDK with Epic Games' UDK and DLLBind

For questions/comments, email cjuncker@logitech.com or lbyrne@logitech.com .