

## HOW TO CREATE A UNITY WRAPPER DLL FOR D-BOX

Since Unity is in C# and our SDK is in C++, we must create a wrapper DLL to be compatible with Unity. D-BOX can expose all its functions in this wrapper. See example and source files included.

**MainDLL.cpp:** The exported functions are exposed here. They must match their C# equivalent.

**DbxStructureDefs.h:** The data, events and structures that will be coming from the Unity application. They must match their C# equivalent.

**DboxSdkWrapper.h:** Declaration of the wrapper's functions that are called by the Main.

**DboxSdkWrapper.cpp:** Definitions of the functions.

**DboxProbe.cs:** The corresponding C# script that matches the DLLs methods and data. Should be placed in the Plugins folder in the Unity Assets.

**Example.cs:** A small example on how to call D-BOX events from the wrapper DLL.

Unity C# called method → C# to C++ through Wrapper DLL Plugin → Wrapper DLL → D-BOX Motion Code DLL

In our experience, D-BOX has been placing both DLLs in the same folder as the Unity Editor executable (installation folder). When the project is built, we place the DLLs in the same folder as the application's executable. This works well and quickly but there are cleaner ways to do so using the Plugins folder.

For more information on creating a C++ plugin for Unity refer to their documentation on native plugins.

Please note the source files included are examples. They can and should be adapted to your project or company's conventions.