CYBERITH SDK

Native C# – Integration Guideline

Cyberith GmbH
Teslastraße 6
3100 St. Pölten
Austria
FN 410899p

For any questions, please contact
Cyberith Support
support@cyberith.com
+43 1 890 17 13

Table of Contents

Prerequisites

Window 7 or newer

Visual Studio

Recommended: Visual Studio Community 2017
https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx

Setup

Dynamic Library

The CybSDK is a dynamic library called **CybSDK.dll.**
In Visual Studio right click your projects **References** and choose **Add Reference…** In the opening wizard choose **Browse** and click **Browse…** After choosing the **CybSDK.dll** from the file system click **OK** to finish the wizard.

Native Dynamic Libraries

CybSDK uses two native C++ libraries located in **/x86/CybSDK_Native.dll** and **/x64/CybSDK_Native.dll.** All that is left to do is to make sure these are places next to your executable in their respective folders.
In Visual Studio open the project properties and navigate to **Build Events** and add the following lines to **Post-build event command line.**

```
xcopy /y /e "$(ProjectDir)\[Path]\CybSDK\*.dll" "$(TargetDir)"
```

SDK Documentation

For full documentation of the C# SDK take a look into the online Doxygen Documentation.
All classes and functions are documented via the XML documentation file **CybSDK.xml** and should show up in your Visual Studio IntelliSense.

Example usage

```
using System;
using CybSDK;

namespace CybSDK_CS_Demo
{
   class CybSDK_CS_ConnectionDemo
   {
      static void Main(string[] args)
      {
         ushort version;
         try
         {
            version = Virt.GetNativeSDKVersion();
         }
         catch (TypeInitializationException e)
         {
            // CybSDK_Native.dll could not be loaded
            Console.Error.WriteLine("[Fatal] Couldn't load CybSDK_Native.dll!");
            Console.ReadLine();
            Environment.Exit(-1);
         }

         IVirtDevice device = Virt.FindDevice();
         if (device == null)
         {
            Console.Error.WriteLine("[Fatal] No Virtualizer connected!");
            Console.ReadLine();
            Environment.Exit(-2);
         }

         VirtDeviceInfo info = device.GetDeviceInfo();
         string product_name = info.ProductName;

         if (!device.Open())
         {
            Console.Error.WriteLine("[Fatal] Unable to connect to Virtualizer!");
            Console.ReadLine();
            Environment.Exit(-3);
         }

         float ring_height = device.GetPlayerHeight();
         float ring_angle = device.GetPlayerOrientation() * 360;
         float movement_direction = device.GetMovementDirection() * 180;
         float movement_speed = device.GetMovementSpeed();

         Console.WriteLine("Connection successful!");
         Console.ReadLine();
      }
   }
}
```