

AWE

Intelligent Headset User Guide



Requirements.

Unity 5.3.x or higher
Jabra Intelligent Headset

For Android builds:

Compatible Android device (Android versions 4 - 6)

(check with [IHS Compatible](#) and [BLE Central, Peripheral Check](#) everything should be supported, but even this is unfortunately not certain for Android devices still)

For iOS builds:

Apple Mac
XCode 7.1 or higher
iOS Device with iOS7.0 or higher

Content and Setup.

In this package you will find everything you need to set up and get started working with, Jabra's Intelligent Headset in Unity. The AWE-IHS plugin included makes it possible to use both iOS and Android devices with your Unity IHS projects. Here follows a description of the important files in this Zip:

unity/Release/AweIHS.unitypackage

You can start a new Unity project and import the "AweIHS.unitypackage" to start from scratch. If you start a new project and want to build for iOS, please follow section.

unity/Release/AWE-IHS-Plugin-Example.apk

AWE-IHS-Plugin-Example.apk is a built Android App that you can transfer to your Android device and install. With this you can test if your Android device will work with Intelligent Headset. If it does connect without problems, then your device is compatible. Make sure to be connected to the IHS device with Bluetooth before starting the app.

unity/

This unity project functions as a starting template for working with IHS and Unity. It also contains a working Xcode project. When building for iOS, make sure to "Append" to update the Xcode project rather than replacing it. Please read the "Additional Guide for building for iOS in Xcode" below.

unity/Assets/Scripts/AweIHS.cs

This c# script contains the core functionality of communicating with the IHS device. Please do not alter this file and always have this file in your project. If you need to start a connection with a IHS device, make another script and call AweIHS.Setup();. If you need sensor data once the AweIHS.ConnectionState() returns "Connected", you can write for example: AweIHS.FusedHeading(); to get the current FusedHeading of the IHS device.

unity/Assets/Scripts/IHSExample.cs

This file has everything already set up to start a connection with a IHS device, and receive all available sensor data from the device. You can modify this file if you like and use it as a template.

unity/Assets/Plugins/

Files in this folder makes sure to setup the connection to IHS. Please do not alter these files and make sure to always have these files in your project. The IHS Android SDK 2.0.0 is already in this folder, so if you are building for Android you do not need to provide the IHS Android SDK.

unity/AWE IHS Plugin Example.unity

This scene runs the IHSExample.cs file and represents all available sensor data from the IHS device. You can modify this scene and use it as a template if you like.

Tips for the Intelligent Headset.

- Before connecting to the IHS, make sure to not have the audio mini-jack cable plugged in.
- When in Bluetooth searching mode, make sure to select the “GN IH a5.0.0” device - not the “IHXXXXXD”.
- Make sure the headset has enough battery. This can only be seen on iOS devices in the upper bar on the BT icon.
- Be careful with the device as it can fall off your head fairly easy, and be aware that it doesn't like rain either :)
- When searching for GPS signal go outside in an open field. Getting a GPS fix can take anywhere between 10 seconds and 10 minutes.
- For optimized performance, audio quality and lower latency, we recommend to attach the audio mini-jack cable between the IHS and your smartphone after it is connected.



Additional guide for building for iOS in Xcode.

“Replace” from Unity to a new Xcode Project

1. Export the Unity project

Build the project for iOS. Save (or “Replace”) from Unity to Xcode - When the Xcode has not previously been setup - then open up the XCode project.

2. Get the IHS plugin

Download from <https://developer.jabra.com/intelligent-headset/> and move the IHS.framework folder into the Xcode project generated by Unity.

3. Implement the plugin

Drag the IHS 2.20 framework into the xcode project folder first and place it in the root (often the folder is called “Xcode”). Add the newly placed IHS framework into Xcode under “Frameworks”. Make sure to “Add to target” and check your project (not the tests though). Then probably add this framework under build settings if Xcode compiler still can’t find “IHS.h”. When opening the IHS framework in Xcode it should contain a folder “Headers” which contains among others “Ihs.h”. Also note, that when handling the download and moving of the framework, always do this with OSX - not Windows, as Windows apparently alters the framework into a regular folder.

4. Within the XCode project make sure the following frameworks are present under General -> “Linked Frameworks and Libraries” - if not add them.

- AudioToolbox.framework
- AVFoundation.framework
- CFNetwork.framework
- CoreLocation.framework
- CoreMotion.framework
- CoreBluetooth.framework
- ExternalAccessory.framework
- MobileCoreServices.framework
- SystemConfiguration.framework
- libc++.tbd

In the info.plist add [com.gn.ihs.public](#) for the key UISupportedExternalAccessoryProtocols. Also add [App Transport Security Settings \(NSAppTransportSecurity\)](#) to allow for headset firmware updates over the air.

5. Possibly add Developer Provision Profile. Also go to Build settings -> Code Signing and change all versions to your team name.

6. Do a clean: Cmd + Shift + K.

7. Do a total clean After this: Cmd + Alt + Shift + K and press Clean.

8. Press Cmd + R, to run the Unity project on the iOS device.

[continued on page 5 >>](#)

“Append” from Unity to Xcode, When the Xcode has already been setup.

1. Delete the reference to the IHS Framework in the Project Navigator sidebar.
2. Add the IHS Framework folder again, by locating it in the project’s Xcode folder. Drag it into the Framework group folder in the Project Navigator. A popup window appears, in which you must make sure the “Add to targets” had the project selected.
3. Do a clean: Cmd + Shift + K.
4. Do a total clean: Cmd + Alt + Shift + K and press Clean.
5. Hit the Play icon, or Cmd+B to just build it but not deploy it.
6. If an error appears regarding LaunchScreen, just clean the project and try again.

<< please feel free to contact us if you require any further information: support@theawe.dk >>