



**Omni SDK**  
**Unity Documentation**

**Version 2.3**

# Table of Contents

<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1 DEFINITION .....	1
1.2 SPECIFICATION OBJECTIVES .....	1
1.3 REQUIREMENTS.....	1
1.4 KNOWN ISSUES .....	1
<b>2. QUICK START GUIDE .....</b>	<b>2</b>
2.1 UPGRADING FROM A PREVIOUS SDK VERSION .....	2
2.2 IMPORTING THE SDK.....	2
2.3 IMPLEMENTING OMNI MOVEMENT .....	3
2.4 USING THE SDK WITH A CUSTOM PLAYER CONTROLLER .....	3
2.5 REGISTER YOUR OMNI FOR DEVELOPMENT .....	3
2.6 MOVEMENT MODES AND OMNI CALIBRATION.....	4
2.6.1 <i>Calibration Application</i> .....	4
<b>3. OMNI SYSTEM OVERVIEW.....</b>	<b>5</b>
3.1 SYSTEM DEFINITION .....	5
3.2 VIRTUIX OMNI HARDWARE .....	5
3.2.1 <i>The Omni</i> .....	5
3.2.2 <i>The Omni Tracking Pods</i> .....	5
3.2.3 <i>The Omni Harness</i> .....	5
3.2.4 <i>The Omni Shoes</i> .....	6
<b>4. OMNI SDK FUNCTIONALITY .....</b>	<b>7</b>
4.1 ADDING STEAMVR CAMERA.....	7
4.2 OMNI MOVEMENT COMPONENT .....	7
4.2.1 <i>Max Speed</i> .....	8
4.2.2 <i>Strafe Speed Multiplier</i> .....	8
4.2.3 <i>Backwards Speed Multiplier</i> .....	8
4.2.4 <i>Gravity Multiplier</i> .....	8
4.2.5 <i>Developer Mode</i> .....	8
4.2.6 <i>Camera Reference</i> .....	8
4.2.7 <i>Dummy Object</i> .....	8
4.3 OMNI AND HMD CALIBRATION.....	8
4.3.1 <i>Calibration Application</i> .....	9
4.3.2 <i>Default / Head-Guided Movement Modes</i> .....	9

## **1. Introduction**

The Omni SDK is used to integrate Omni Input into your Unity application. We believe that the Omni is a key component to creating truly immersive VR games with natural locomotion, and we are excited to see how you choose to use our hardware. We hope you enjoy the experience that the Omni provides.

### **1.1 Definition**

The Omni SDK Unity Documentation describes the procedure to get Omni Input working in a Unity application.

### **1.2 Specification Objectives**

The Omni SDK has the following objectives:

1. To provide the necessary steps for a Unity application to use the Omni as an input device.
2. To outline the various functions that can be called through the Omni Plugin.
3. To give a basic understanding of how the Omni Hardware functions.

### **1.3 Requirements**

These are the requirements for using the SDK:

1. The Omni SDK is currently designed to only work with the HTC Vive.

### **1.4 Known Issues**

N/A

## 2. Quick Start Guide

The Quick Start Guide outlines the essential steps to integrate the Omni SDK into your game.

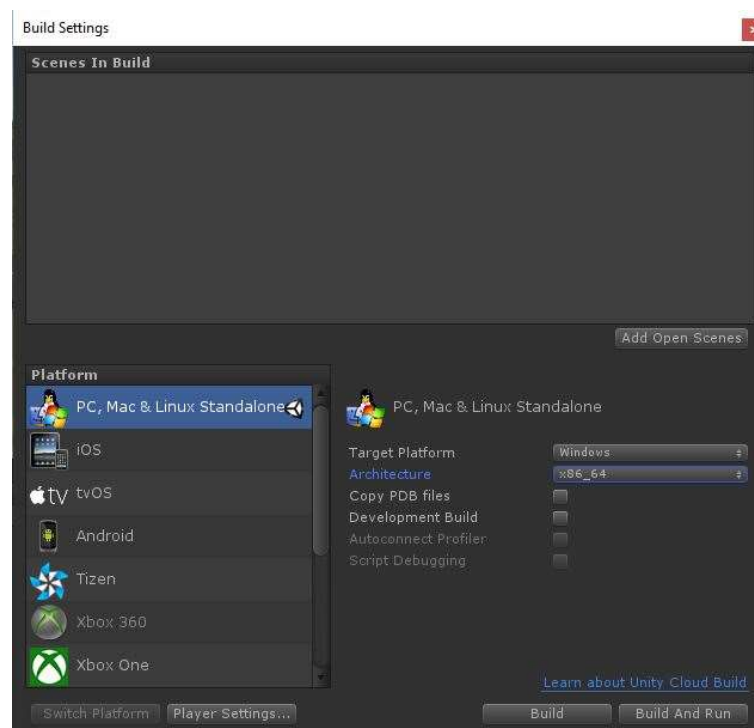
### 2.1 Upgrading from a Previous SDK Version

When upgrading from an older version of the SDK, delete the old SDK in your project before proceeding to section 2.2. Do not attempt to over-write the SDK in your project.

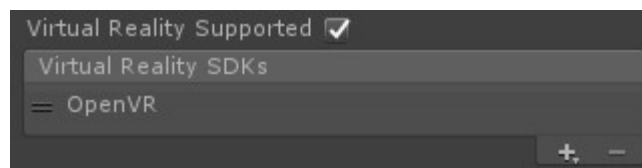
### 2.2 Importing the SDK

This section outlines the steps to properly import the Omni SDK into your Unity project.

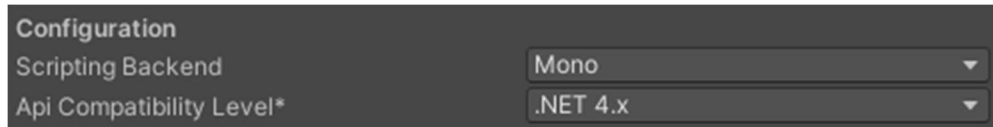
1. Set up your Omni and connect it to your computer.
2. Install and run Omni Connect.
3. In your Unity Project, under build settings, set your project to **x86\_64**.



4. Unless working in Developer Mode (where no VR HMD needs to be installed), ensure that your HTC Vive is properly connected.  
**NOTE:** For more information about Developer Mode, see section 4.2.5.
5. Import the latest SteamVR plugin from the Unity Asset Store.
  - a. After importing the plugin, go to **Edit > Project Settings > Player**. Verify that Virtual Reality Supported is checked and the OpenVR is listed under the Virtual Reality SDKs.



6. In your Unity project, under **Edit > Project Settings > Player**, find "API Compatibility Level" under "Other Settings" and change it from ".NET Standard 2.0" to ".NET 4.x". Failing to set this parameter to ".NET 4.x" will cause errors in the Omni SDK after import.



7. Import the Omni SDK package into your project.
  - a. To import the SDK, choose **Assets > Import Package > Custom Package**.
  - b. Navigate to where the downloaded package is, click it, and select import.
  - c. After the package finishes decompressing, select Import in the bottom right.

## 2.3 Implementing Omni Movement

1. You can open the "OmniTestScene" to test Omni movement in this scene or place the "OmniCharacterController" in any of your scenes to test it there.
  - a. For more details on the parameters for the characters, continue to section 4.2.

## 2.4 Using the SDK with a Custom Player Controller

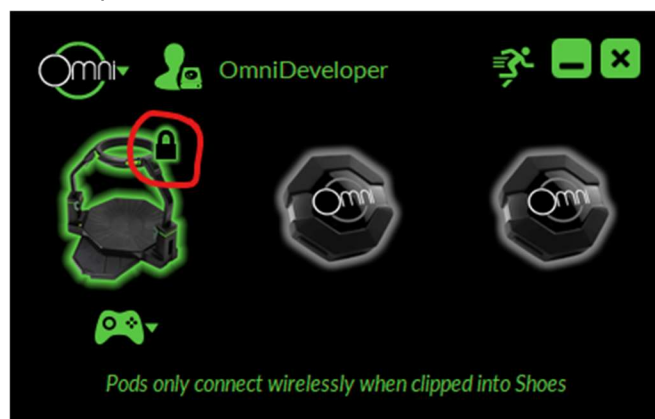
This section outlines how to add SDK functionality to a custom player controller instead of using the Omni Character Controller.

1. Put the "Omni Movement Component" on your custom player character. This component converts raw input from the Omni into a movement vector that can be used to move a character.
2. Use the movement vectors from the "Omni Movement Component" to move your character controller. (Look at OmniController\_Example for a reference on how to do this)

## 2.5 Register Your Omni for Development

This section outlines the process for registering your Omni for development. This step is only necessary if your Omni is encrypted.

1. Install Omni Connect on your system (found in the SDK package).
2. Open Omni Connect and connect your Omni to your PC via USB.
3. Power on your Omni and look at the Omni Icon in Omni Connect. If there is a lock icon in the top right corner, then you have an encrypted Omni and must register the device with your Omni provider (this is the case for most Omni owners). If there is no lock icon, then you can skip this step and move on.



4. To register your device, contact your Omni provider and provide the serial number for any Omnis that you wish to use for development.

## 2.6 Movement Modes and Omni Calibration

The Omni user can choose between two movement modes: “Default Movement” and “Head-Guided Movement”.

### Default Movement:

In the Omni’s default movement mode, players move in the direction that their waist/harness is facing, independent of their looking direction. Sensors in the ring of the Omni read the rotation of the harness, which determines the player’s walking orientation in-game. This allows players to look left and right while walking forward, without changing direction.

### Head-Guided Movement:

In this movement mode, players move in the direction that their headset is facing. The rotation of the player’s body in the Omni is not considered. This mode is easier for novice players or for games that require fast and sharp turns.

The Omni user can switch between the Omni’s default movement mode and the “Head-Guided Movement” mode through the Omni calibration application.

### 2.6.1 Calibration Application

To use the Omni in the default movement mode, which uses the Omni Harness rotation to calculate the character’s forward direction instead of using the HMD forward direction, the application must have access to the relative rotation offset between the Omni and the HMD. Both the Omni and the HMD use a different *local* coordinate system. When the yaw rotation of each component is facing the same direction in the real world, they are not necessarily facing the same direction in the game world. This offset between the two coordinate systems will cause the character to move in a perceivably incorrect direction in the game world in the default movement mode. To correct the movement, the two coordinate systems must be calibrated so that when the directions of both systems are aligned in the real world, they are also aligned in the game world.

To fix this offset, the Omni must be calibrated to the HMD tracking space. The Omni is calibrated using an external application which is included in the SDK package. It is also available to all customers. This application saves an “Omni Yaw Offset” value that specifies the difference between the Omni coordinate system and the HMD tracking space coordinate system.

Once the Omni has been calibrated, the application will work in the default movement mode automatically without any additional development requirements. The Omni Movement Component handles the offset value from the calibration application. The Omni will only need to be recalibrated if the Omni is physically moved, or if the SteamVR room setup changes.

Please use the external Omni calibration application included in the SDK package to configure and test your application in the default movement mode.

### 3. Omni System Overview

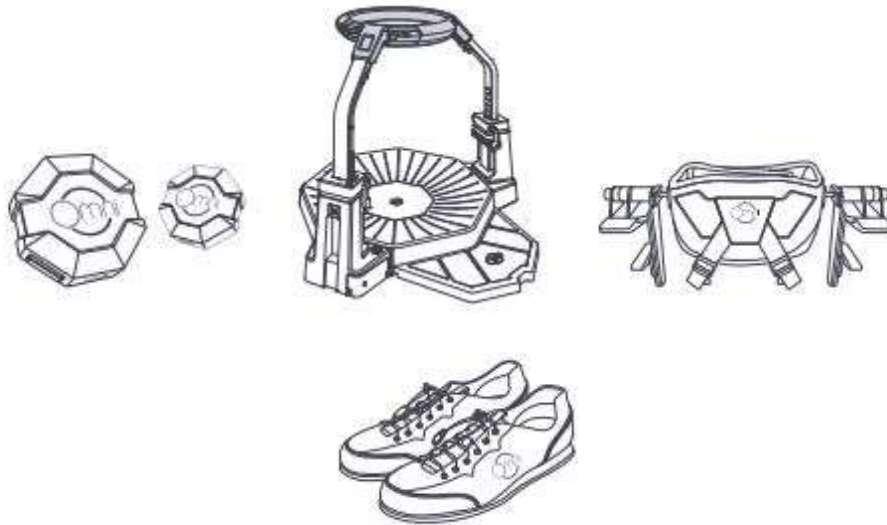
This section provides a high-level explanation of what the Omni is and how the device functions.

#### 3.1 System Definition

The Virtuix Omni™ is a first-of-its-kind active virtual reality motion platform. Active VR, where your actions in the virtual world are controlled by first-person navigation like walking or running, creates an unprecedented sense of immersion that cannot be experienced sitting down. The Virtuix Omni™ lets you walk, run, sit, and strafe with 360-degree freedom of movement, allowing you to control your avatar without restraints.

#### 3.2 Virtuix Omni Hardware

The Omni system comprises four pieces of hardware that you must understand before developing a game for the Omni: the Omni itself, the Omni Tracking Pods, the Omni Harness, and the Omni Shoes.



##### 3.2.1 The Omni

The Omni is an omnidirectional treadmill that allows players to walk, run, sit, and strafe with 360-degree freedom of movement, allowing you to control your avatar without restraints. The key component that enables this movement functionality is the Omni's concave, low-friction base. This base, combined with the low-friction shoes that the player wears, defines the environment in which the player can walk and run.

##### 3.2.2 The Omni Tracking Pods

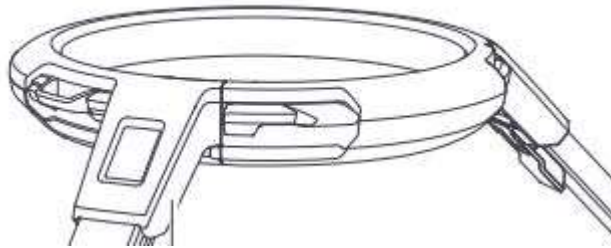
The Omni Tracking Pods are attached to the Omni Shoes and contain IMUs (Inertial Measurement Units) that measure the rotation and acceleration of the player's feet. Our motion library uses this data in conjunction with the Omni ring data (see below) to infer how the player is moving on the Omni: forward, backwards, sideways, and at what speed.

##### 3.2.3 The Omni Harness

The Omni Harness has two main functions: safety and tracking.

*Safety:* The Omni Harness securely holds the player upright in the Omni, preventing him or her from falling while inside the ring.

*Tracking:* The Omni Ring contains sensors that read both the position and rotation of the Omni Harness. This data is used to infer whether the player is walking forwards in the front area of the ring or walking backwards in the rear area of the ring. The rotation of the Harness is also measured, providing torso orientation data that enables the default movement mode (moving and looking in separate directions).



### **3.2.4 The Omni Shoes**

The Omni Shoes feature a proprietary low-friction sole material and a sole configuration that stabilize the foot and enable a natural gait when using the Omni. Omni Shoes are required to use the Virtuix Omni.



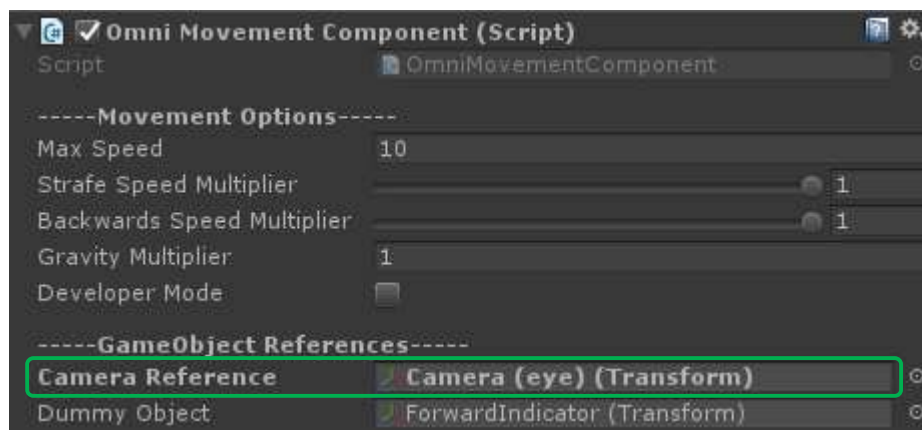
## 4. Omni SDK Functionality

This section explains the components of the Omni SDK and how to use them for your project.

### 4.1 Adding SteamVR Camera

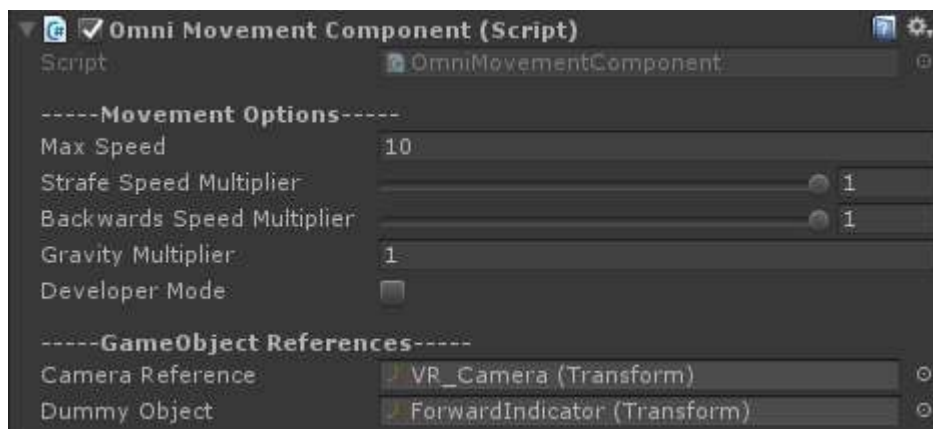
If you want to use the SteamVR **[CameraRig]** prefab with the “Omni Character Controller”, which will allow you to use the Vive hand controllers as well as some other SteamVR features, perform the following steps:

1. Import the SteamVR package from the Unity Asset Store (**Window > Asset Store**).
2. In the Unity Project Tab, go to the **Assets > SteamVR > Prefabs** folder, and find the **[CameraRig]** prefab.
3. Add the **[CameraRig]** prefab as a child to the “CameraSetupRootOffset” on the “Omni Character Controller” prefab
4. Remove the “VR\_Camera” from the “Omni Character Controller” prefab.
5. On the “Omni Character Controller”, find the “Camera Reference” variable on the “Omni Movement Component”. Assign the “Camera (eye)” from the **[Camera Rig]** to this variable.



### 4.2 Omni Movement Component

This component is used to convert input from the Omni to a movement vector that can be used to move the player.



### 4.2.1 Max Speed

This speed variable represents the speed at which the character moves in-game when the player is running at full speed on the Omni. The Omni reports input values ranging from -1 to 1, with “1” being the input value when the player runs at maximum speed on the Omni. If the player runs at half of maximum speed, then the Omni Input is “.5”, and the character will run in-game at a speed equal to half of this speed variable.

The default max speed is set to 10. You can increase this value if you want the player to move faster in-game and decrease if you want the player to move slower. You may want to adjust this value depending on the size of the playable space, and the speed of gameplay.

### 4.2.2 Strafe Speed Multiplier

This parameter multiplies the speed of strafing on the Omni. To fine-tune the strafing speed up and down, adjust this variable.

### 4.2.3 Backwards Speed Multiplier

This parameter multiplies the speed of backwards movement on the Omni. To fine-tune the backwards speed up and down, adjust this variable. We use a value of 0.4 in our games.

### 4.2.4 Gravity Multiplier

This parameter multiplies the gravity on the character. Leave the value at 1 for a normal gravity value. Decrease this value if you want the player to fall at a slower speed.

### 4.2.5 Developer Mode

Developer Mode allows you to use the setup with a keyboard and mouse and without the need for an Omni or HMD. This developer mode is useful when integrating the Omni SDK at your desk away from an Omni unit. Movement is done with WASD and looking with the mouse. To use the Omni with an HMD, you must disable this mode. Also disable this mode before creating a build for the Omni.

### 4.2.6 Camera Reference

This parameter is a reference to the camera on the player.

### 4.2.7 Dummy Object

This parameter is a reference to a dummy object on the player that is used to calculate the forward direction of the player.

## 4.3 Omni and HMD Calibration

To use the Omni in the default movement mode, which uses the Omni Harness rotation to calculate the character’s forward direction instead of using the HMD forward direction, the application must have access to the relative rotation offset between the Omni and the HMD. Both the Omni and the HMD use a different *local* coordinate system. When the yaw rotation of each component is facing the same direction in the real world, they are not necessarily facing the same direction in the game world. This offset between the two coordinate systems will cause the character to move in a perceivably incorrect direction in the game world in. To correct the movement, the two coordinate systems must be calibrated so that when the directions of both systems are aligned in the real world, they are also aligned in the game world.

To fix this offset, the Omni must be calibrated to the HMD tracking space.

### **4.3.1 Calibration Application**

The Omni is calibrated using an external application which is included in the SDK package. It will also be available to all customers. This application saves an “Omni Yaw Offset” value that specifies the difference between the Omni coordinate system and the HMD tracking space coordinate system.

Once the Omni has been calibrated, the game should work with the default movement mode automatically without any additional development requirements. The Omni Movement Component already handles the offset value from the calibration application. The Omni will only need to be recalibrated if the Omni is physically moved, or if the SteamVR room setup changes.

The calibration application will be used by customers to change between the default movement mode and the “Head-Guided Movement” mode, and to calibrate the Omni if in the default movement mode.

### **4.3.2 Default / Head-Guided Movement Modes**

#### **Default Movement:**

In the Omni’s default movement mode, players move in the direction that their waist/harness is facing, independent of their looking direction. Sensors in the ring of the Omni read the rotation of the harness, which determines the player’s walking orientation in-game. This allows players to look left and right while walking forward, without changing direction.

#### **Head-Guided Movement:**

In this movement mode, players move in the direction that their headset is facing. The rotation of the player’s body in the Omni is not considered. This mode is easier for novice players or for games that require fast and sharp turns.