

# How to adapt your AirConsole Unity game for AndroidTV

---

An AirConsole app for AndroidTV is available on the Google Play store (<https://play.google.com/store/apps/details?id=com.airconsole.androidtv&hl=en>). Unfortunately, the AndroidTV cannot run the WebGL versions of Unity games that we have in the AirConsole store. In order to run your Unity-made AirConsole games on Android TV, you need to create a native Android Build instead of the regular WebGL build. Some adjustments may have to be made to your game project, as detailed in this guide.

When you switch platforms to Android and export your game to an AndroidTV, the AirConsole Plugin creates a webview that shows AirConsole content within your AndroidTV App.

Following this guide will give you an APK file which can then be uploaded to the Google Play Store. If you are thinking about porting your AirConsole game to AndroidTV, please contact us beforehand.

## Tools

- Make sure your Unity Version is 5.4.1f1 or higher
- Make sure your AirConsole Plugin version is 1.4 or higher
- Download the Android SDK and JDK from within the Unity Editor
- Go to **Unity** → **Preferences** → **External Tools** and download SDK and JDK using the download buttons available.

## Platform

Before you start working on your AndroidTV Port, it's important that you have uploaded a .zip of your WebGL build to [airconsole.com/developers](http://airconsole.com/developers) (<http://airconsole.com/developers>) and create the Game ID which we will need for the next steps. Read the "Publish your game" guide which explains how to upload your game.

In **File** → **Build Settings**, choose 'Android'. Reimporting assets for the platform might take a moment.

# BundleID

Go to **Edit** → **Project Settings** → **Player**.

In the Other Settings tab, you'll find the parameter Bundle Identifier. For your AndroidTV build to work, this needs to be identical with the Game ID that you've given your game on [airconsole.com/developers](http://airconsole.com/developers) (<http://airconsole.com/developers>).



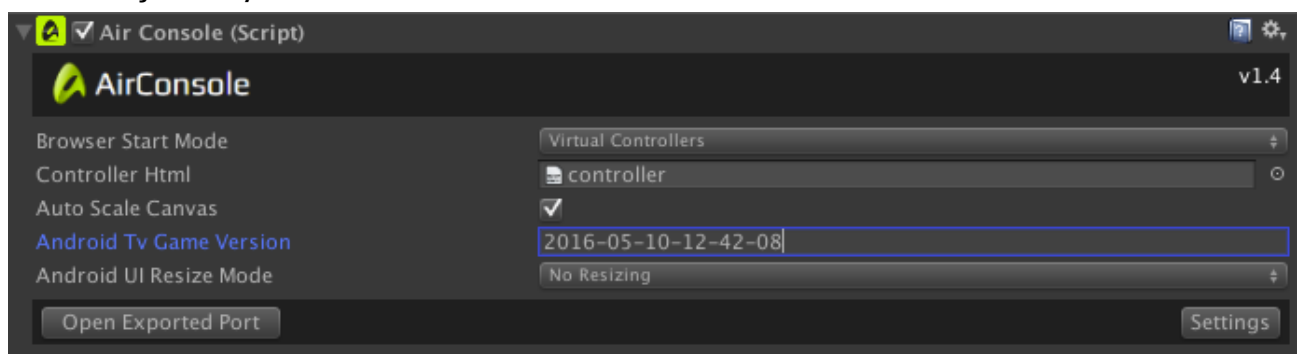
## Screen and Controller Data

In order for your exported game to be able to access the controller and all necessary JavaScript and image files, you need to:

1. Have a WebGL build with all needed screen and controller Data uploaded to [airconsole.com/developers](http://airconsole.com/developers) (<http://airconsole.com/developers>)
2. Copy the Game upload version number of your game.

**Game upload version 2016-05-10-12-42-08:**

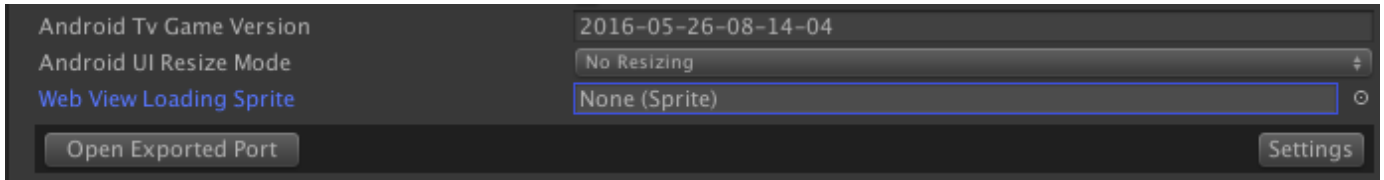
3. Paste the number into the **Android TV Game Version** field on the AirConsole GameObject in your Scene.



4. Save your scene and create an Android build. The **gameName.apk** file you get can be deployed to your AndroidTV device

## Loading Screen Sprite

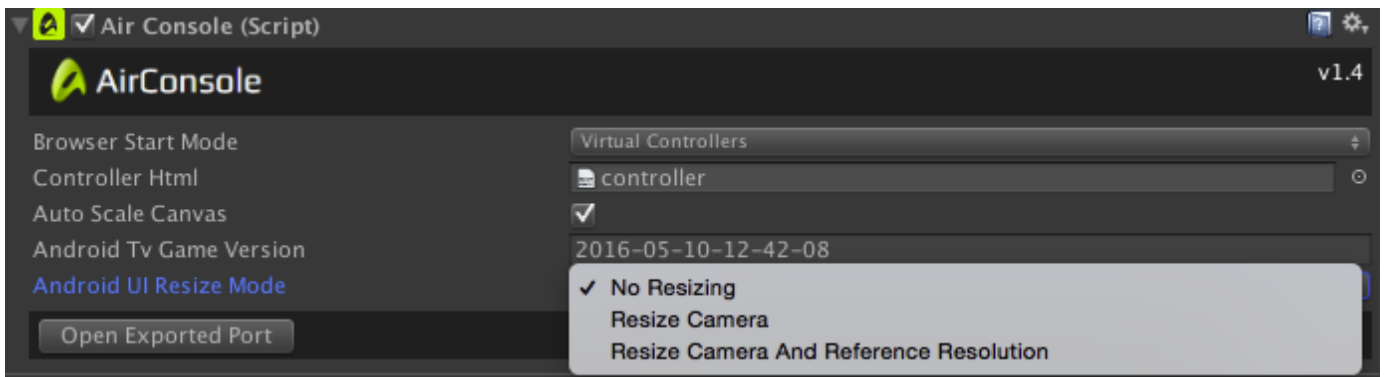
You can choose a custom loading screen logo to be displayed when your game starts on Android TV. If there is none, a default AirConsole logo will be displayed.



## UI

On AndroidTV, the AirConsole Default UI bar will be laid over the game. The Default UI shows the AirConsole Logo, the session code and connected players' portraits and connection statuses.

To make sure your UI is displayed correctly despite the cutoff at the top, the AirConsole Object offers three options on how to deal with Canvases.



- **No Resizing**

Your UI is left as it is, but the Default UI is laid over. If you choose this option you can either:

- Manually adjust your UI so that nothing important will be cut off at the top
- Hide the default UI using `AirConsole.instance.ShowDefaultUI(false)` and display the session code yourself as part of your ingame UI

- **Resize Camera**

The height of the default UI is subtracted from the MainCamera's `pixelRect`. On your Canvas, set **RenderMode** to **Screen Space - Camera**. Now, if your UI elements' `RectTransforms` are correctly anchored, they will not be cut off by the default UI. Resizing the Camera will change its aspect ratio.

- **Resize Camera and Reference Resolution**

The camera is resized as described in option 2). In addition, the Reference Resolution of your Canvas Scalers is adjusted to keep the aspect ratio it had before. This will ensure that UI elements are placed correctly in relation to the 3D elements in your game scene. It means, however, that your canvas is narrower than your Camera.

Note: The resizing happens in the OnLevelWasLoaded event. Do not depend on the order of OnLevelWasLoaded being before or after the OnEnable/Awake/Start Events.

---

AirConsole.com (<http://www.airconsole.com>) | @ N-Dream AG (<http://www.n-dream.com>)  
| Blog (<http://blog.airconsole.com/>)

Follow @AirConsole