

Advanced Framework - EOS Link Documentation

1

HumanCodeable

May 6, 2021

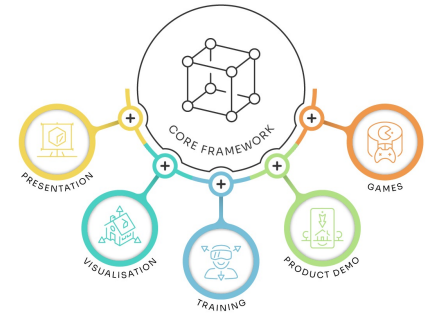


CONTENTS

1	Introduction	2
2	Preparations	2
2.1	Epic Games Developer Account	2
2.2	Creating a Product	2
2.3	Setting Up Epic Account Services	3
2.4	P2P Setup	3
2.5	Checking your Product Details	3
2.6	Vivox Setup	3
3	Purchases	3
3.1	Advanced Framework Core 4.1	4
3.2	EOS Core Plugin	4
3.3	Vivox Core Plugin	4
3.4	EOSLink Utility	4
4	Configuration	5
4.1	Out of Engine Preparation	5
4.2	Enable Plugins	5
4.3	Engine Settings	5
4.4	Product Settings	6
5	Testing	6
6	Appendix	8
6.1	References	8
6.2	DefaultEngine.ini - Manual Guide	8
6.3	DefaultEngine.ini - Manual Guide	8

1 INTRODUCTION

EOSLink is a free Advanced Framework Utility to connect to Epic’s On-line Services and enable simple multiplayer solutions. It connects to the Advanced Framework Core 4.1 and all extensions or utilities based on it via the code *AFUEOSLink*. The philosophy behind EOSLink is to provide AF customers with a simple, out-of-the-box solution for reliable multiplayer networking of AF-based projects. While the Epic Online Services provide a wide variety of multiplayer features such as game services logins, parties and achievements, the EOSLink Utility specifically does not implement these features in order to streamline the multiplayer experience in just 2 clicks, all from within VR or in a desktop or mobile environment.



ATTENTION

The EOS Link multiplayer solution requires a third party purchases in addition to at least the Advanced Framework Core 4.1.

2 PREPARATIONS

The largest amount of work to implement this module is related to setting up your organization’s information inside the Epic Games Developer Portal. The key steps to having a functional EOSLink Utility are described below, but it is recommended that you thoroughly set up your organization’s information inside the developer portal.

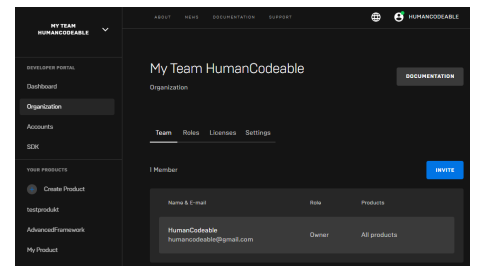


Figure 1. After all steps of section 2.1.

2.1 Epic Games Developer Account

To use the EOSLink Utility you need an epic games developer account which is going to handle your product and online policies.

Where to go: <https://dev.epicgames.com/>

What to do:

- 1) **Registration** - create a developer account
- 2) **Necessary Licences** - accept the *EPIC Standard Services* and *Account Services* licenses
- 3) **Organization** - set up your organization details
- 4) **Team** - enter yourself as team member with owner role and all other team members with their respective roles

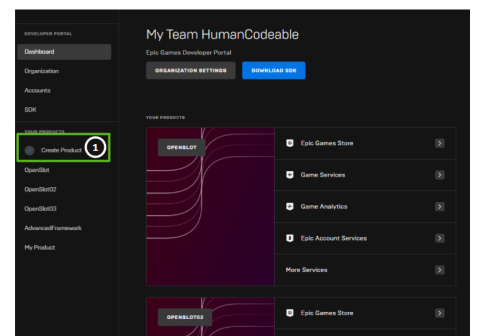


Figure 2. Dashboard shown the products.

2.2 Creating a Product

The product basically represents your application in the your epic games developers account.

Where to go: the dashboard of your account

What To Do:

- 1) **Create Product** - hit the create product button shown in Figure 2
- 2) **Product Name** - enter the name of your application
- 3) **Cover Image** - upload an image meeting the specifications given in the dialog

💡 If you dont have a fitting image at hand you can skip this step.

2.3 Setting Up Epic Account Services

The epic account services are just a part of the services epic games provides for you product. However, this is not a documentation on how to use the developer account, so we stick to the bare minimum necessary to set up a working multiplayer application using EOS Link and the Advanced Framework.

Where To Go: the epic account services settings of your product

What To Do:

- 1) **Getting There** - hit the highlighted button in Figure 3
- 2) **Start Configuration** - hit the *Configure* button highlighted in Figure 3
- 3) **Brand Settings** - enter at least the required information on your application
 - **Application Name** - the name your application is known as (required)
 - **Application Logo** - upload the logo of your application
 - **Privacy Policy URL** - enter a full web URL which contains your privacy policy (required)
 - **Support** - enter a web page or email address your clients can look at for support
- 4) **Permissions** - set all permission requirements to *enabled*
- 5) **Clients** - create a client either for peer2peer (see section 2.4) or dedicated server (see section 6.3)

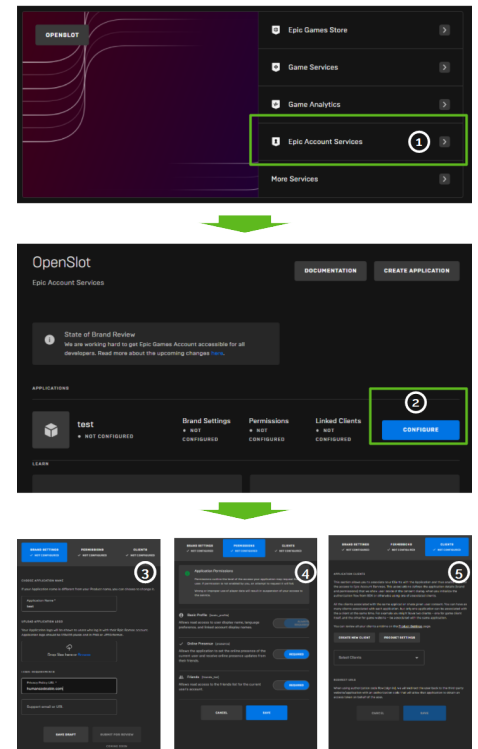


Figure 3. All steps of section 2.3.

2.4 P2P Setup

Peer-to-peer (P2P) is the client policy supporting a listening server which hosts the game as well as allows a player to play like a client.

Where To Go: Clients tab of your epic games online services setup

What To Do:

- 1) **Getting Started** - Hit the *Create New Client* button
- 2) **Client Name** - fill in the client name
- 3) **Client Policy** - hit the *Add New Policy* button
 - **Client Policy Name** - fill in the name *PolicyP2P*
 - **Client Policy Type** - set it to *Peer2Peer*
- 4) **Save** - hit the save button

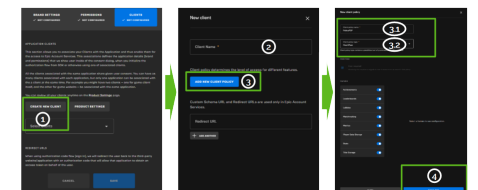


Figure 4. All steps of section 2.4.

💡 You can use any other name that suits your application instead

2.5 Checking your Product Details

You can check out your product details under product settings in the sidebar. The client policies you find under the clients tab.

2.6 Vivox Setup

3 PURCHASES

After setting up the epic games developer account and the product everything is ready to install the plugin and incorporate it into your Advanced Framework Project. As mentioned above this feature relies on the separate purchase of at least 2 packages from the unreal marketplace as well as

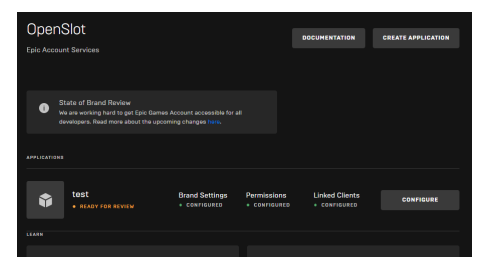


Figure 5. After all steps of section 2.4.

download a free Advanced Framework Utility. Purchase and install all packages and plugins marked as necessary before progressing to the next section.

3.1 Advanced Framework Core 4.1

The Advanced Framework Core provides your project with a basis of functions, classes and other entities which allow you to create your application more swiftly. The AF Core comes with a variety of examples including a multiplayer lobby which you can use for your first multiplayer setup.

Creator: HumanCodeable

Link: <https://unrealengine.com/marketplace/en-US/product/advanced-vr-framework>

Status: Necessary Purchase

3.2 EOS Core Plugin

The EOS Core Plugin serves to introduce the epic online services (EOS) into your unreal project.

Creator: eelDev

Link: <https://unrealengine.com/marketplace/en-US/product/eoscore>

Status: Necessary Purchase

3.3 Vivox Core Plugin

The Vivox Core connects your unreal project to the vivox voice chat system, so you can use it for for your multiplayer.

Creator: eelDev

Link: <https://unrealengine.com/marketplace/en-US/product/vivoxcore>

Status: Optional Purchase

⚠ If you forgo the Vivox Core Plugin, you have to set up voice chat differently

WARNING

Currently, the EOSLink Utility expects the Vivox Core Plugin. If you want to forgo Vivox you need to delete all functions related to Vivox from your version of the EOSLink Utility.

3.4 EOSLink Utility

The EOSLink Utility is a free Advanced Framework Utility that provides you with the necessary functionalities to use the EOS Core Plugin with your Advanced Framework Core project

Creator: HumanCodeable

Link: http://ansgarjahn.de/Downloads/Public/AFUEOSLink_1.0.zip

Status: Necessary Free Advanced Framework Utility

WARNING

The EOSLink Utility is not a standalone product. It cannot be used without at least the Advanced Framework Core. It is also not downward compatible to versions below the Advanced Framework Core 4.1.

4 CONFIGURATION

This section describes how you need to configure your project to successfully create a multiplayer application using the Advanced Framework and the epic games online services plugin. Make sure you have at least a copy of the Advanced Framework Core 4.1 and of the Advanced Framework EOSLink Utility in your unreal engine content folder.

RECOMMENDATION

Create a new Advanced Framework project to understand how the EOSLink module operates. Its integration in existing projects, while not overly complicated, is beyond the scope of this documentation. Moreover, you will have significantly less problems integrating this into existing projects once you acquired working knowledge of the module in a test project.

4.1 Out of Engine Preparation

To prepare the project for the new Plugins etc. the configurations of your unreal engine need a few adjustments.

Where To Go: Unreal Projects/Advanced Framework/Config

What To Do:

- 1) **DefaultEngine.ini** - The EOSLink Utility comes with a modified DefaultEngine.ini file. You find it in the folder Unreal Projects/Advanced Framework/Content/AFUEOSLink. Use it to replace the original DefaultEngine.ini of your project
- 2) **Android** - In the same folder you find a folder named Android. If you want to be able to build your application on an Android system (which includes the Oculus Quest) copy that folder, too.
- 3) **Alternative Method** - if you prefer to modify your DefaultEngine.ini file manually follow the instructions in the appendix (section 6.2)

- - - Start the Unreal Engine now - - -

WARNING

Make sure your Unreal Editor is not running during this step!

4.2 Enable Plugins

The EOS Core and the Vivox Core are entered to your project as plugins that you need to enable before you can use them.

Where To Go: Unreal Engine Plugins Browser Tab

What To Do:

- 1) **Find Plugins** - go to the tab eelDev Plugins. There you should find at least the EOS Core Plugin
- 2) **Enable Plugins** - enable the EOS Core and Vivox Core Plugins

4.3 Engine Settings

With the new plugins in place your engine requires a few settings modifications.

Where To Go: Unreal Engine Settings Browser Tab

What To Do:

- 1) **Maps And Modes** - enter the following settings here:
 - **Default Map:** Map_Multiplayer_Lobby
 - **Game Instance Class:** BP_GameInstance_Main_EOSLink
- 2) **Asset Manager** - perform the following modifications
 - **Map Primary Asset Type** - add the directory /Game/AFUEOSLink/Maps
- 3) **Packaging** - Find the *List of maps to include in a packaged build* and add the Map_Multiplayer_Lobby to it.

💡 You need to enter the full file pathway of the map here.

4.4 Product Settings

To use the epic games online services your engine also needs the product settings of the product you created in your epic games developer account.

Where To Go: Project Settings/Game/EOS Core Plugin Tab

What To Do:

- 1) **Epic Games Account** - open your project and go the *General* tab of the project settings
- 2) **EOS Core Settings** - copy the following settings into the field with the same name:
 - Product ID
 - Sandbox ID
 - Deployment ID
 - Client ID
 - Client Secret

💡 If you modified your Default-Engine.ini file by hand you can skip this step and restart the unreal engine right away.

- - - Restart the Unreal Engine now - - -

5 TESTING

If you followed the setup up until you should be ready to try out the multiplayer. So just go to the Map_Multiplayer_Lobby and hit play.

💡 We recommend testing multiplayer in the standalone game mode.

NOTE

You need at least 2 computers both with the setup described above to test in multiplayer. Alternatively you can build your application and distribute the build to the computers you want to use for testing.

Once you successfully entered the multiplayer lobby press the start button on the pallet. Now you have the following possibilities:

- **Host a Session** - press the *Host* button. The session is created and added to the session list automatically
- **Refresh the Sessions List** - press the curly arrow button to refresh the sessions list if you hosted a session and it is not displayed yet
- **Join a Session** - select the session of your choice to join. Your player pawn is spawned automatically at one of the starting positions for a joined session.

WARNING

The whole instructions up until here are focusing on a peer-2-peer or listening server setup. For a dedicated server a few additional steps are necessary in the setup as well as in the testing part, which are added in the Appendix (section 6.3).

6 APPENDIX

6.1 References

6.2 DefaultEngine.ini - Manual Guide

6.3 DefaultEngine.ini - Manual Guide