

# Cross-Buy asset for Unity

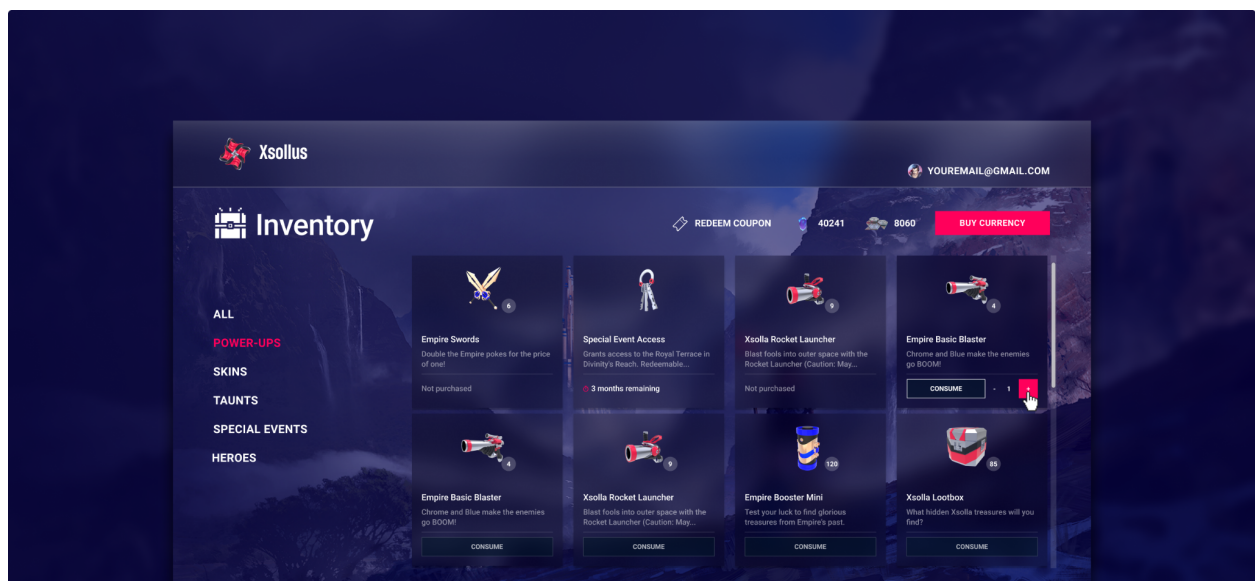
Game Commerce asset is a set of classes and methods that you can integrate into your Unity app to work with Xsolla products.

After integration of the asset, you can use:

- [Login](#) to authenticate users and manage the friend system and user account
- [In-Game Store](#) to manage player's inventory

The asset is suitable if your application is published on a platform that cannot use Xsolla solutions for payments and in-game store for some reason.

[Try our demo to learn more.](#)



## Note:

Cross-Buy asset is already included as part of the [Game Commerce](#) asset. You can download the Game Commerce if you need a broader set of features.

Cross-Buy asset includes [Login & Account System](#) asset.

Do **NOT** install these assets separately.

For a better understanding of which Xsolla asset to choose, use the table:

	Game Commerce asset	Login & Account System asset	Cross-Buy asset
In-game store			
Virtual currency	+		
Virtual items	+		

Player inventory	+		+
Bundles	+		
Promotional campaigns	+		
<b>Login</b>			
Authentication	+	+	+
User management	+	+	+
<b>Payment UI</b>	+		
<b>Additional features</b>			
UI builder	+	+	+
Battle pass	+		

## Requirements

### System requirements

- 64-bit OS
- Windows 7 SP1 and later
- macOS 10.12 and later
- The version of Unity not earlier than 2019.4.19f1

### Target OS

- Android
- macOS
- Windows 64-bit

Additionally, the asset supports [creating WebGL build](#) to run your application in a browser.

#### Note:

We recommend you use the Mono compiler for desktop platforms as it's compatible with the provided in-game browser. If you use other browser solutions, you can use the IL2CPP compiler instead. To create game builds for Android, you can use either Mono or IL2CPP compilers.

# Integration

Before you integrate the asset, you need to sign up to [Publisher Account](#) and set up a new project.

More instructions are on the [Xsolla Developers portal](#).

## Usage

Xsolla provides APIs to work with its products. The Game Commerce asset provides classes and methods for API calls, so you won't need to write boilerplate code. Use the [tutorials](#) to learn how you can use the [asset methods](#).

## Known issues

### Conflict of multiple precompiled assemblies with Newtonsoft.json.dll

The issue appears when importing the asset on Unity version 2020.3.10f1 and later. The following error message is displayed:

```
Multiple precompiled assemblies with the same name
Newtonsoft.json.dll included on the current platform. Only one
assembly with the same name is allowed per platform.
```

The conflict arises because the `Newtonsoft.json.dll` library is included in both the Unity Editor and the asset. The library is included in the versions 2020.3.10f1 and later of the editor. And the asset includes the library to support the earlier versions of Unity Editor.

**Issue status:** Fixed in 0.6.4.4.

### Newtonsoft.json.dll could not be found

The problem appears if you upgraded a pre-existing project to Unity version 2020.3.10f1 and later. Importing an asset from the [Unity Asset Store](#) into such a project is accompanied by many error messages like this:

```
The type or namespace name 'Newtonsoft' could not be found (are you
missing a using directive or an assembly reference?)
```

The problem occurs because the `Newtonsoft.json.dll` library is not included in the asset for Unity version 2020.3.10f1 and later. As part of the editor, the library is supplied for versions 2020.3.10f1 and later, but when updating the project for these versions, the library requires manual installation.

**Issue status:** Fixed in 0.6.4.4.

# Unable to resolve reference UnityEditor.iOS.Extensions.Xcode

## Issue description

The issue appears when using External Dependency Manager on Unity version 2020.1.0f1 and later.

When building the application, an error message is displayed:

Assembly

'Packages/com.google.external-dependency-manager/ExternalDependencyManager/Editor/Google.IOSResolver\_v1.2.161.dll' will not be loaded due to errors:

Unable to resolve reference 'UnityEditor.iOS.Extensions.Xcode'. Is the assembly missing or incompatible with the current platform? Reference validation can be disabled in the Plugin Inspector.

**Issue status:** Fixed in 0.6.4.5.

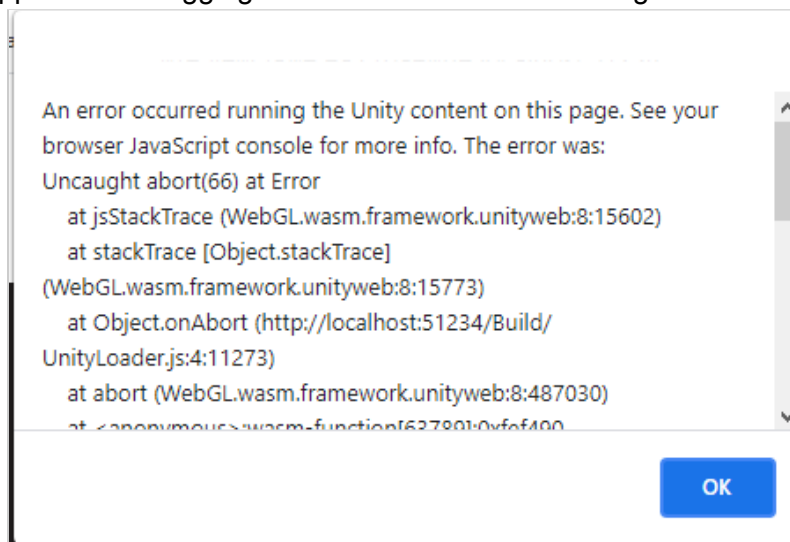
## Workaround

Install iOS Build Support module from Unity Hub.

# Error occurred running Unity content on page of WebGL build

## Issue description

The issue may appear when logging in WebGL build. The following error message is displayed:



See details on cause of the issue on [Unity Issue Tracker](#).

**Issue status:** Won't fix.

## Workaround

1. Open Unity project.

2. Click **Edit > Project Settings** in the main menu.
3. In the **Player** section, go to the WebGL build settings tab.
4. Go to the **Other Settings** section.
5. Uncheck **Strip engine code** box.
6. Go to the **Publishing Settings** section.
7. Check the **Decompression Fallback** box.
8. Create a new WebGL build.

## Legal info

[Explore legal information](#) that helps you work with Xsolla.

Xsolla offers the necessary tools to help you build and grow your gaming business, including personalized support at every stage. The terms of payment are determined by the contract that can be signed via Publisher Account.

**The cost of using all Xsolla products is 5% of the amount you receive for the sale of the game and in-game goods via Xsolla Pay Station.** If you do not use Xsolla Pay Station in your application, but use other products, contact your Account Manager to clarify the terms and conditions.

---

## License

See the [LICENSE](#) file.

## Community

[Join our Discord server](#) and connect with the Xsolla team and developers who use Xsolla products.

## Additional resources

- [Xsolla official website](#)
- [Developers documentation](#)
- [Code reference](#)
- API reference:
  - [Login API](#)
  - [Commerce API](#)