Pure Pool Documentation

Troubleshooting - View Webpage

Contents

- Troubleshooting
- Managed code stripping
- Meta Files - Missing Script
- 'IPunPrefabPool' could not be found
- ArgumentException: The Assembly UnityEditor is referenced by Umbrace.Unity.PurePool
- ArgumentException: The manager must be set to load from Resources to use it with Photon. Set UseResources to true.
- error CS0117: 'PhotonNetwork' does not contain a definition for 'UsePrefabCache'
- Can not play a disabled audio source

Troubleshooting

Unfortunately, even the best software sometimes has undiscovered issues, or behaves in unexpected ways. If you need help, please check the sections below to see if your issue has already been addressed. If you don't find anything relating to your issue, feel free to contact us or write a post in our support forum.

Managed code stripping

Unity supports a feature called managed code stripping, which removes unused code from the build. This helps reduce the size of the final program, and is especially important for builds using IL2CPP to speed up the build time.

However, managed code stripping can sometimes fail to detect that certain code will be used at runtime, particularly in the case of C# reflection. If it incorrectly removes code that's needed, you may experience errors or crashes.

To ensure Pure Pool is not stripped and your code will always work no matter what stripping level you use, you may find it useful to add instructions to prevent Unity from performing code stripping on it. To do this, you should create a file called link.xml inside your Assets folder. There can be more than one, and they can be inside any subfolder, so you could put it in the Pure Pool folder.

The link.xml file should look like this:

```
<linker>
  <!--Preserve an entire assembly-->
  <assembly fullname="Umbrace.Unity.PurePool" preserve="all" />
  <assembly fullname="Umbrace.Unity.PurePool.PlayMaker" preserve="all"
/>
</linker>
```

Meta Files - Missing Script

When using any external libraries (DLL files) with Unity, there is the possibility for GUID-related issues to occur. These issues manifest as "*Missing (Mono Script)*" messages in the inspector, and messages in the console along the lines of "*The referenced script on this Behaviour is missing!*".

Read how to fix this issue.

'IPunPrefabPool' could not be found

When using the Photon Unity Networking (PUN) integration script for Pure Pool, you may encounter the following error message:

> Assets/Plugins/Umbrace.Unity.PurePool/Integration Libraries/Photon Unity Networking/PrefabPool.cs(25,28): error CS0246: The type or namespace name `IPunPrefabPool' could not be found. Are you missing an assembly reference?

This error indicates that the IPunPrefabPool interface from PUN cannot be found. In most cases, the fix is simply to move the PrefabPool.cs integration script out of the Plugins folder, to another location in your Assets folder. In addition, please ensure you have the latest version of Photon Unity Networking installed.

ArgumentException: The Assembly UnityEditor is referenced by Umbrace.Unity.PurePool

When you try to build your Unity project for one of the available Unity players, you may encounter the following error message:

> ArgumentException: The Assembly UnityEditor is referenced by Umbrace.Unity.PurePool ('Assets/Plugins/Umbrace.Unity.PurePool/Umbrace.Unity.PurePool.dll').
>
> But the dll is not allowed to be included or could not be found.

This error indicates that you are using an older version of Pure Pool (version 1.5 or older) and are using the **Debug (Editor)** binaries of Pure Pool, which are the default binaries installed from the Asset Store, but they are only suitable for use inside the Unity editor. To build your project, you should instead use either the **Debug** or the **Release** binaries, which can be found inside the **Pure Pool Binaries.zip** archive, in the Assets\Plugins\Umbrace.Unity.PurePool\Binaries folder.

Simply extract the appropriate binaries to your Assets\Plugins\Umbrace.Unity.PurePool folder, following the Installation guide if you have any doubts.

Alternatively, **update to version 1.6** or newer, where the **Debug (Editor)** binaries have been removed, and both **Debug** and **Release** configurations can be used to build your project.

ArgumentException: The manager must be set to load from Resources to use it with Photon. Set UseResources to true.

When running your Unity project with Photon Unity Networking integration, you may encounter the following error message:

> ArgumentException: The manager must be set to load from Resources to use it with Photon. Set UseResources to true.
>
> Umbrace.Unity.PurePool.Photon.PrefabPool.set_Manager

This error indicates that the `NamedGameObjectPoolManager` being assigned to `PrefabPool.Manager` has its `UseResources` property set to false. The UseResources property must be set to true to assign the Manager property. This is required as most of the GameObjects being spawned by PUN will not have been set up in the NamedGameObjectPoolManager manually, and should instead be loaded from the Resources folders.

To fix this, simply set the UseResources property of the NamedGameObjectPoolManager to true before assigning the manager:

```
// Set the manager to load by name from the Resources folders.
// This is required as most objects spawned by PUN will not be set up
in the manager already, and will need to be loaded.
NamedGameObjectPoolManager.Instance.UseResources = true;

// Assign the manager, so it knows how to acquire and release from the
pools.
```

```
this.punPrefabPool.Manager = NamedGameObjectPoolManager.Instance;
```

error CS0117: 'PhotonNetwork' does not contain a definition for 'UsePrefabCache'

When running your Unity project with Photon Unity Networking integration, you may encounter the following error message:

> error CS0117: 'PhotonNetwork' does not contain a definition for 'UsePrefabCache'

This error indicates that your project is using PUN2, but the integration library for PUN Classic has been imported. You should instead use the integration library for PUN2.

To fix this, delete the `PrefabPool.cs` file from your Assets folder, and then import `Assets\Plugins\Umbrace.Unity.PurePool\Integration Libraries\Photon Integration Library V2.unitypackage`.

Can not play a disabled audio source

When running your Unity project with Photon Unity Networking integration, you may encounter the following error message:

> Can not play a disabled audio source
>
> UnityEngine.AudioSource:Play()

This error indicates that you have a component using the notification mode to run code whenever an object is acquired from the pool. The code in the Acquire method is attempting to play an AudioSource, but the object is disabled.

This occurs because PUN2 requires that objects acquired from the pool are disabled, so any Pure Pool code that runs on Acquire will happen while the object is still disabled.

To fix this, you should instead move such code from the Acquire method to either the Unity OnEnable method, or to the PUN OnPhotonInstantiate method (see IPunCallbacks for PUN Classic and IPunInstantiateMagicCallback for PUN2).