

DevResources

OddBitOut - Version 1.0.0

DevResources

[Introduction](#)

[Editor Tools](#)

[API](#)

[DevResources.FindObjectsOfTypeAll](#)

[DevResources.Load](#)

[DevResources.LoadAll](#)

[DevResources.LoadAsync](#)

[DevResources.UnloadAsset](#)

[DevResources.UnloadUnusedAssets](#)

[DevResources.ResetCache](#)

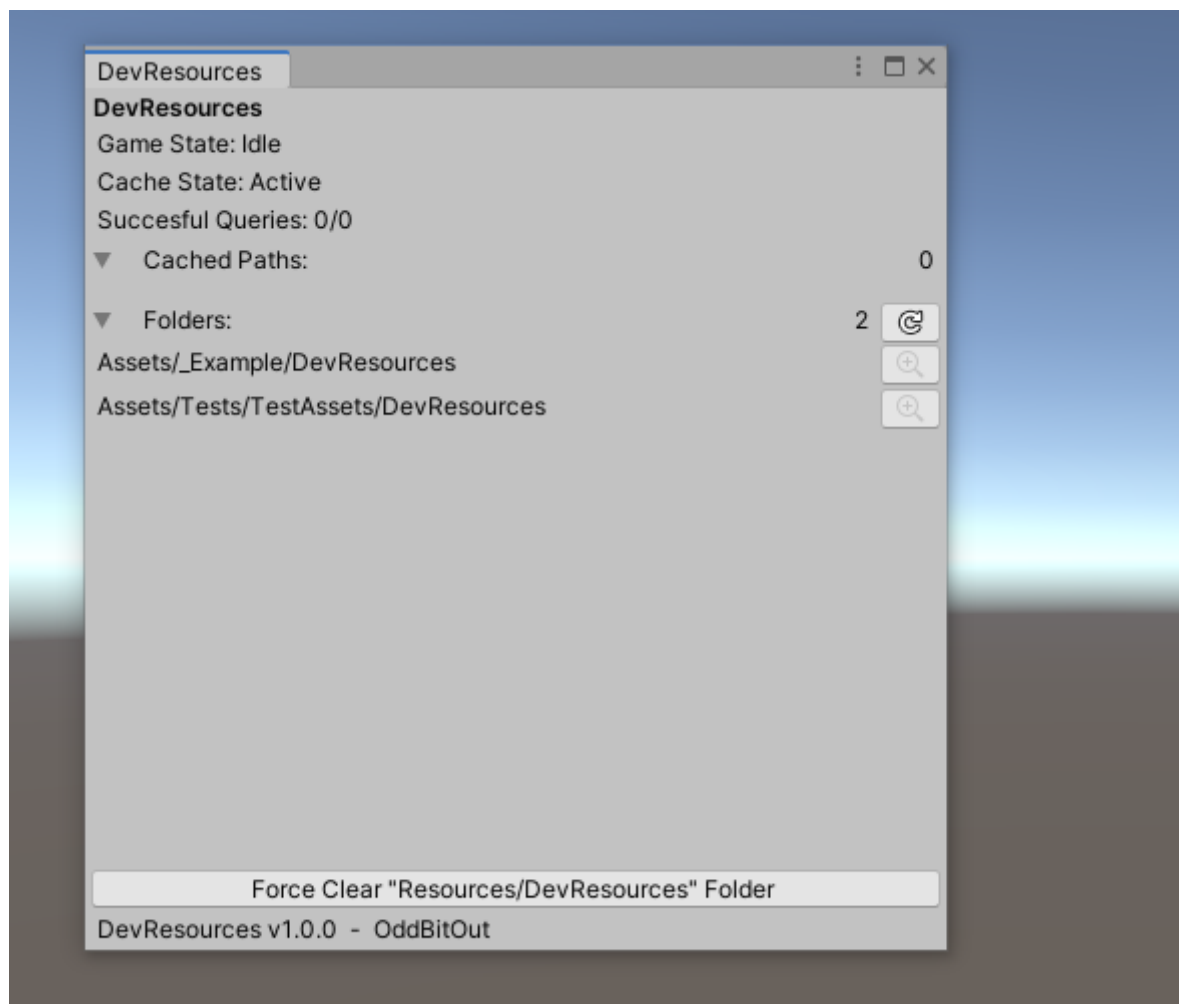
Introduction

Description: DevResources allows you to separate your development resources from release assets without having to try and find them all throughout the project. Simply create a DevResources folder anywhere in your project, drag and drop your files in and use it the same way you would use Resources .Load(...) normally.

- Allows multiple DevResources folders throughout the project.
- Works across all platforms when creating a development build.
- Strips out assets on release builds.
- Full source available.

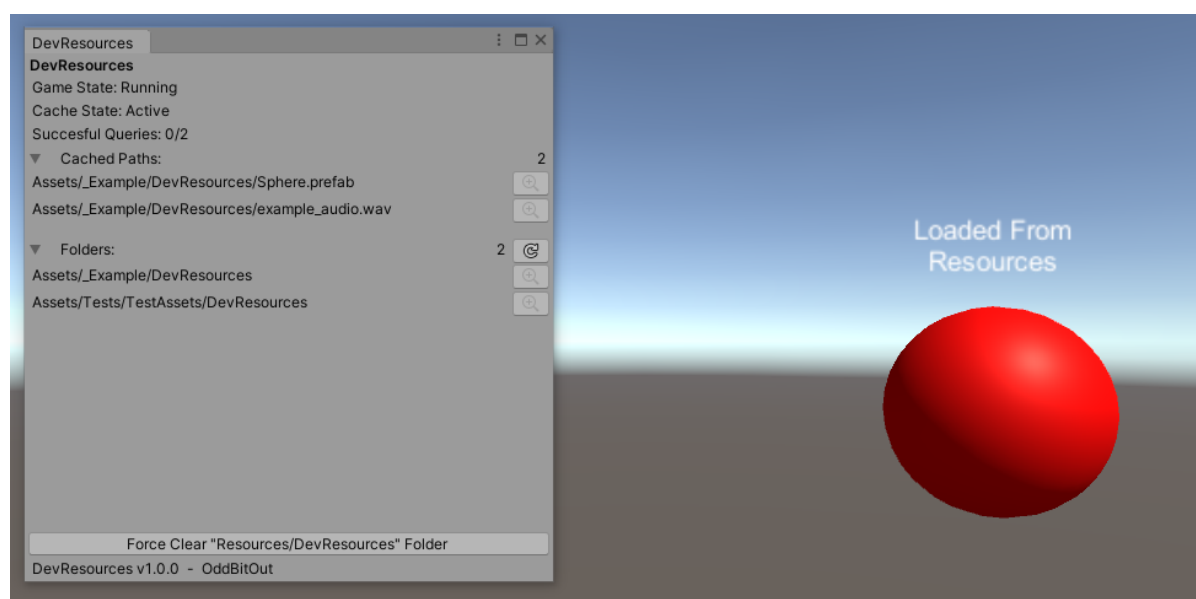
Editor Tools

You can load up the DevResources tool via 'Tools/DevResources' from the Editor menu.



This tool will give you a quick view of all the DevResources that you have in your project, you can setup multiple folders in your project and they will be amalgamated into a single folder when building an executable version of your game.

When you build an executable version of your game a PreBuild step will copy the contents of the various folders around your project and a PostBuild step will clear the folder. (In the event that the folder is not cleared after building you can use the 'Force Clear' button to clear the folder).



While running the game you can see the list of object paths that have been cached by the system while running the game.

API

DevResources.FindObjectsOfTypeAll

```
public static Object[] FindObjectsOfTypeAll(Type type)
```

```
public static TType[] FindObjectsOfTypeAll<TType>() where TType : Object
```

Based on: <https://docs.unity3d.com/ScriptReference/Resources.FindObjectsOfTypeAll.html>

| Parameters | |
|------------|--|
| type | The type of object to find in the DevResources folder. |

Finds and return all object of a given type contained within any of the DevResources folder.

```
// Find all the loaded objects of a given type
using OddBitOut.DevResources;

public class DevResourcesExample : MonoBehaviour
{
    List<GameObject> GetSceneObjectsNonGeneric()
    {
        List<GameObject> objectsInScene = new List<GameObject>();

        foreach(GameObject g in DevResources.FindObjectsOfType<GameObject>())
        {
            objectsInScene.Add(g);
        }

        return objectsInScene;
    }
}
```

DevResources.Load

```
public static TType Load<TType>(string path) where TType : Object
```

```
public static Object Load(string path)
```

```
public static Object Load(string path, Type systemTypeInstance)
```

Based on: <https://docs.unity3d.com/ScriptReference/Resources.Load.html>

| Parameters | |
|--------------------|---|
| path | Pathname of the target folder. When using the empty string (i.e., ""), the function will load the entire contents of the DevResources folder. |
| systemTypeInstance | Type filter for objects returned. |

Loads an object given a path relative to any DevResources folder within the project. The path does not need to include the DevResources part of the path or the extension of the file.

```
// Load assets from the DevResources folder.
using OddBitOut.DevResources;

public class DevResourcesExample : MonoBehaviour
{
    public void LoadAssets()
    {
        // Load an audio clip at 'Assets/DevResources/example_audio'
        AudioClip clip = DevResources.Load<AudioClip>("example_audio");

        // Load an Object at 'Assets/DebugObjects/DevResources/Sphere'
        Object obj = DevResources.Load("Sphere");
    }
}
```

DevResources.LoadAll

```
public static Object[] LoadAll(string path)
```

```
public static Object[] LoadAll(string path, Type systemTypeInstance)
```

```
public static TType[] LoadAll<TType>(string path) where TType : Object
```

Based on: <https://docs.unity3d.com/ScriptReference/Resources.LoadAll.html>

| Parameters | |
|--------------------|---|
| path | Pathname of the target folder. When using the empty string (i.e., ""), the function will load the entire contents of the DevResources folder. |
| systemTypeInstance | Type filter for the objects returned. |

Loads all assets in a folder or file at a path in a DevResources folder.

If the path refers to a folder all the assets in the folder will be returned. If the path refers to a file, only that asset will be returned.. The path is relative to any DevResources folder inside the Assets folder of your project.

```
// Load all the object of a type
using OddBitOut.DevResources;

public class DevResourcesExample : MonoBehaviour
{
    void Start()
    {
        // Load all textures in 'Assets/Textures/DevResources/DebugTextures'
        var textures = DevResources.LoadAll<Texture2D>("DebugTextures");

        foreach (Texture2D t in textures)
        {
            Debug.Log(t.name);
        }
    }
}
```

DevResources.LoadAsync

```
public static ResourceRequest LoadAsync(string path)
```

```
public static ResourceRequest LoadAsync(string path, Type type)
```

```
public static ResourceRequest LoadAsync<TType>(string path) where TType : Object
```

Based on: <https://docs.unity3d.com/ScriptReference/Resources.LoadAsync.html>

| Parameters | |
|--------------------|--|
| path | Pathname of the target folder. When using the empty string (i.e., ""), the function will load the entire contents of the Resources folder. |
| systemTypeInstance | Type filter for objects returned. |

Asynchronously load an asset stored at a path in the DevResources folder.

Returns a DevResourceRequest from which the asset can be retrieved once the loading operation is completed. Only objects of the type will be returned if this parameter is supplied.

The path is relative to any DevResources folder inside the Assets folder of your project, the extension should be omitted.

DevResources.UnloadAsset

```
public static void UnloadAsset(Object assetToUnload)
```

Based on: <https://docs.unity3d.com/ScriptReference/Resources.UnloadAsset.html>

DevResources.UnloadUnusedAssets

```
public static AsyncOperation UnloadUnusedAssets()
```

Based on: <https://docs.unity3d.com/ScriptReference/Resources.UnloadUnusedAssets.html>

DevResources.ResetCache

```
public static void ResetCache()
```

Clears the DevResources cache of cached asset paths.