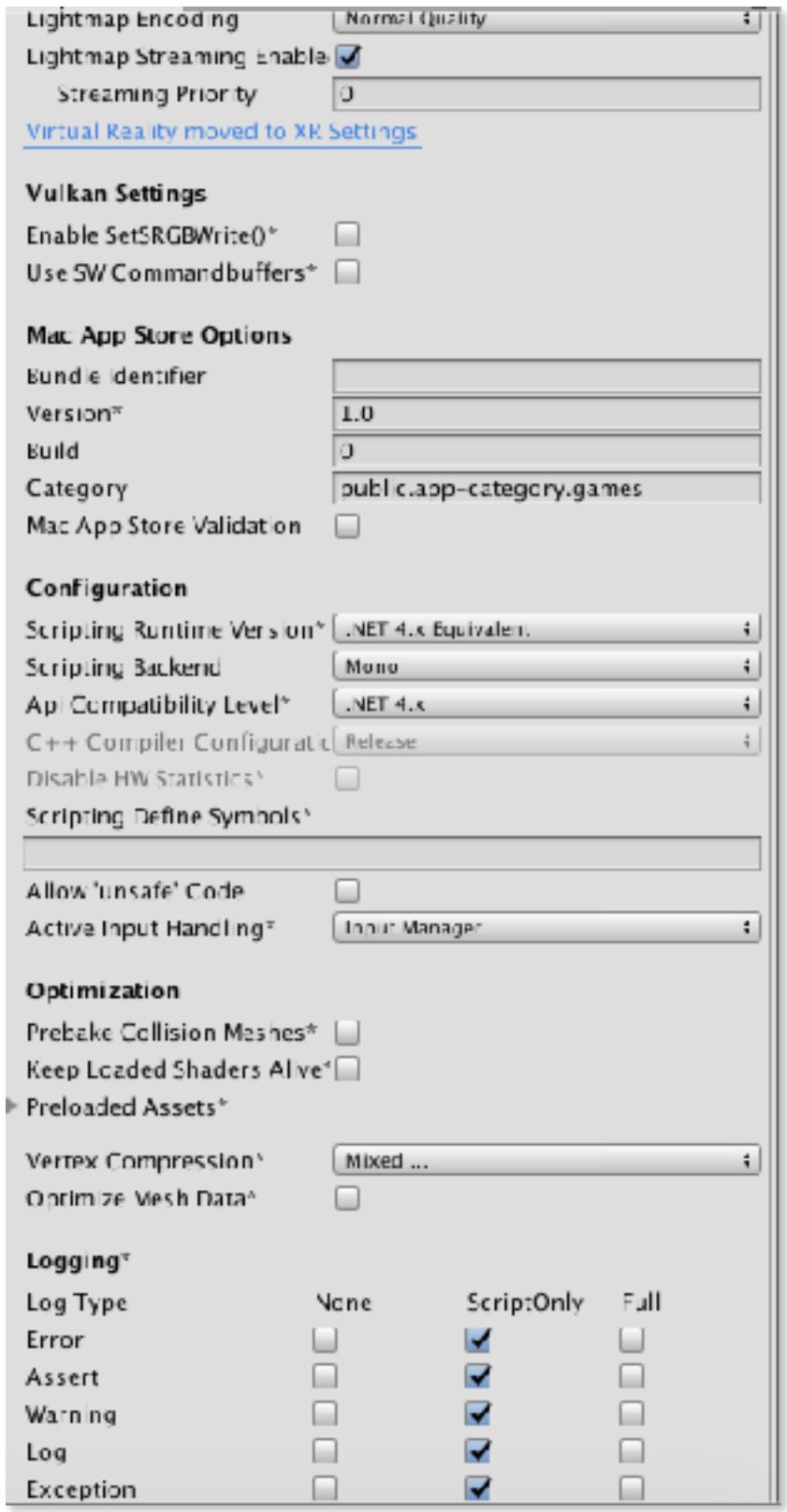
# VP documentation

# Table of contents

# IMPORTANT

THE PLUGIN WORKS ONLY IN .NET 4 SO ACTIVATE THIS OPTION IN UNITY EDITOR. YOU MUST ALSO ADD INCREMENTAL COMPILER PACKAGE BECAUSE THE PLUGIN USE C# 7 SYNTAX.



# WHAT IS IT?

VP was created to help save, visualize and modify data in Unity game engine. With VP the process of saving data is very simplified and fast.

I am always open to hearing new ideas for improvements or suggestions and of any problems that you might encounter while using VP plugin.

You can email me any time at mciissee@gmail.com and I will respond shortly.

# WHAT'S IN THE PACKAGE?

The plugin is fully installed at « Assets/InfinityEngine/Frameworks/VP » with the following structure.

- **InfinityEngine/Scripts**

Contains the assemblies that hold the core code of the plugin.

- **InfinityEngine/Frameworks/VP**

Contains the scripts of the of VP plugin.

- **InfinityEngine/Gen**

Contains resources auto generated by the plugin. You cannot delete this folder and if you delete it, the class VPEditor.Resolver will regenerate it again.

- **VP.cs**

Hold the core code of VP plugin.

- **VPEditor.cs**

Hold the code that creates the editor window that let you visualize edit your preferences. The preferences that you create using the editor are usable at runtime and you can share them between your projects (only the names of the preferences not the values).

- **R2.cs**

This script is regenerated by the plugin each time you edit your preferences using the editor window. It works like android R class. It contains static references to all the preferences. This allows you to avoid using hardcoded strings in your projects and increase your productivity.

# GETTING STARTED

First of all, to use the plugin, you have to import it from the assets store and use the following directive in the head of your scripts :

```
using InfinityEngine.Serialization;
```

- **How to save and load a preference ?**

VP class provides access to static methods which starts with the prefix « Set » like « VP.SetInt(string, int) » or the prefix « Get » like « VP.GetInt(string) ».

Theses methods allow to save and load a preference represented by the class « VP.Pref<T> ».

**Example :**

The following code is an example of how to save and load an integer preference identified by the key « **score** » :

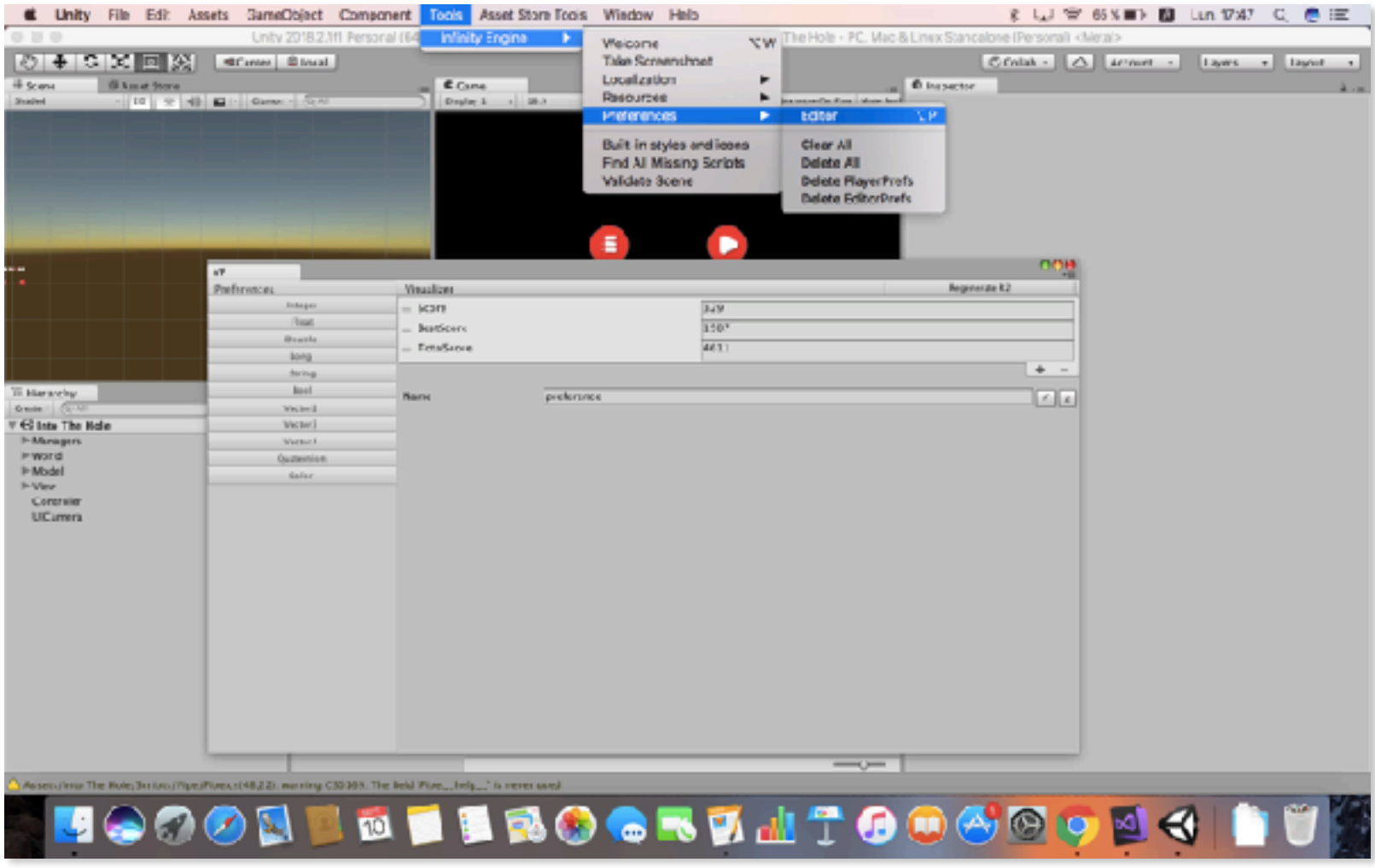
```
VP.SetInt("score", 10); // creates new preference with the key 'score'
VP.Save(); // Write the modification on disk.
Debug.Log(VP.GetInt("score")); // print 10 on the console
```

- How to visualize a preference ?

Open the editor located at « Tools/Infinity Engine/Preferences/Editor ».

1. Use the « Preferences » section to select the type of the preferences that you want to visualize.
2. Use the « Visualizer » section to edit or create your preferences.
3. Click on the button « Regenerate R2 » to regenerate the script « R2 ».

The purpose of the generated script is to avoid you to deal with preferences by using hardcoded strings. If you use hardcoded strings, you could make some mistakes like a spelling mistake.




**Example :**

The following code sets the value of the preference 'score' created in the last example to 100 by using R2 class instead of hardcoded string.

```
// This code call automatically 'SetInt' method of VP class.
R2.score.Value *= 10;
VP.Save(); // Write the modification on disk.
```

- **How to delete a data ?**

1. To delete a key, call the static method « VP.DeleteKey(PrefType, string) ».
2. To delete all keys, call the static method « VP.DeleteAll() ».
3. You can also use the editor to delete a key or all keys at « Tools/InfinityEngine/VP/Editor »

# FAQ

- **Where the data are saved ?**

The data are placed at Application.persistentDataPath + visualPrefs.vp

- **Will this work on all platform ?**

The plugin has not been tested on all unity platforms, it has been tested and works perfectly on the platforms : **Android**, **IOS**, **Standalone** and **WebGL**.
The plugin should work on all other platforms but in case of problem, you can let me know.

- **Can the plugin serialize all types ?**

In the current version, the plugin saves only the types referenced in the enumeration PrefTypes.

- **Is R2 script regenerated after I create a preference during play mode ?**

When you create a preference in script, it is not added to R2 class, so the good way to use this plugin is to create your preferences using the editor then generate R2 class by using the editor.

- **Is the plugin auto save preferences ?**

When you update a preference by calling the setters functions or by using R2 class properties, the plugin does not write the modifications on disk for performance purpose, you must the method VP.Save(). If you edit a preference from the editor, the modifications are saved.

# FINAL WORDS

Thanks you if you purchased this asset, if not you can purchase it at http://u3d.as/GLW.

If you like the asset, please rate it.

Support is available at mciissee@gmail.com.

Make sure you check out my other assets at my assets store page.

## Good luck for your projects.