

Contents

I.	What is ProjectPrefs?	2
II.	Editor	2
III.	Player.....	3
IV.	Ini File Parser	3
V.	Using the Demos	4

I. WHAT IS PROJECTPREFS?

ProjectPrefs is a utility kit for Unity which allows you to share custom project configuration between multiple Unity Editors and access all configuration in-game.

The API is based on UnityEditor.EditorPrefs and UnityEngine.PlayerPrefs classes making the usage very similar to what you are used to do normally in your projects.

II. EDITOR

The editor class allows you to read and write into an .ini configuration file located originally at Assets/Resources/prefs.ini.txt. That way the configurations can easily be read from inside your game.

USAGE

First add this using directive line at the top of your C# script:

```
1 using Bottle.Editor;
```

Now you can make use of ProjectPrefs by accessing its static methods:

```
1 // Declare some variables.
2 private int intValue;
3 private float floatValue;
4 private bool boolValue;
5 private string stringValue;
6 private string multilineValue;
7
8 public void Setup()
9 {
10     // Access the keys contained in your configuration file.
11     // The second parameter is optional and represents
12     // the initial value of the key.
13     intValue = ProjectPrefs.GetInt("Int", 8);
14     floatValue = ProjectPrefs.GetFloat("Float", 2.5f);
15     boolValue = ProjectPrefs.GetBool("Bool", true);
16     stringValue = ProjectPrefs.GetString("String", "Hello, world!");
17     multilineValue = ProjectPrefs.GetString("MultilineString");
18 }
```

After your work is done with your class variables, it's time to save your work back into the configuration file:

```
1 private void SaveData()
2 {
3     // Register any modification you've made in your key/value pairs.
4     ProjectPrefs.SetInt("Int", intValue);
5     ProjectPrefs.SetFloat("Float", floatValue);
6     ProjectPrefs.SetBool("Bool", boolValue);
7     ProjectPrefs.SetString("String", stringValue);
8     ProjectPrefs.SetString("MultilineString", multilineValue);
9
10    // After telling ProjectPrefs which variables were changed
11    // it's time to actually rewrite the file on disk.
12    ProjectPrefs.Save();
13 }
```

III. PLAYER

The player class is a simplified version of the editor class that allows you to your project configurations as a TextAsset located into Assets/Resources folder.

USAGE

First add this using directive line at the top of your C# script:

```
1 using Bottle;
```

Now you can read your ProjectPrefs values in the exact same way you did in the editor class:

```
1 // Declare some variables.
2 private int intValue;
3 private float floatValue;
4 private bool boolValue;
5 private string stringValue;
6 private string multilineValue;
7
8 void Setup()
9 {
10     // Access the keys contained in your configuration file.
11     // The second parameter is optional and represents
12     // the initial value of the key.
13     intValue = ProjectPrefs.GetInt("Int", 8);
14     floatValue = ProjectPrefs.GetFloat("Float", 2.5f);
15     boolValue = ProjectPrefs.GetBool("Bool", true);
16     stringValue = ProjectPrefs.GetString("String", "Hello, world!");
17     multilineValue = ProjectPrefs.GetString("MultilineString");
18 }
```

And use it at your fashion:

```
19 void OnGUI()
20 {
21     GUI.Label(new Rect(10, 10, 80, 20), "Int Value:");
22     GUI.Label(new Rect(90, 10, 100, 20), intValue.ToString());
23
24     GUI.Label(new Rect(10, 30, 80, 20), "Float Value: ");
25     GUI.Label(new Rect(90, 30, 100, 20), floatValue.ToString());
26
27     GUI.Label(new Rect(10, 50, 80, 20), "Bool Value: ");
28     GUI.Label(new Rect(90, 50, 100, 20), boolValue.ToString());
29
30     GUI.Label(new Rect(10, 70, 80, 20), "String Value: ");
31     GUI.Label(new Rect(90, 70, 500, 20), stringValue);
32
33     GUI.Label(new Rect(10, 90, 80, 20), "Multiline Text: ");
34     GUI.Label(new Rect(90, 90, 500, 500), multilineValue);
35 }
```

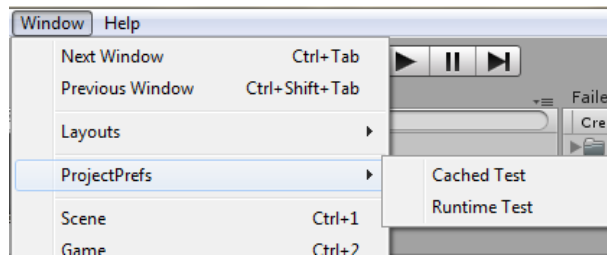
IV. INI FILE PARSER

ProjectPrefs uses Ini File Parser library for reading and writing .ini files.

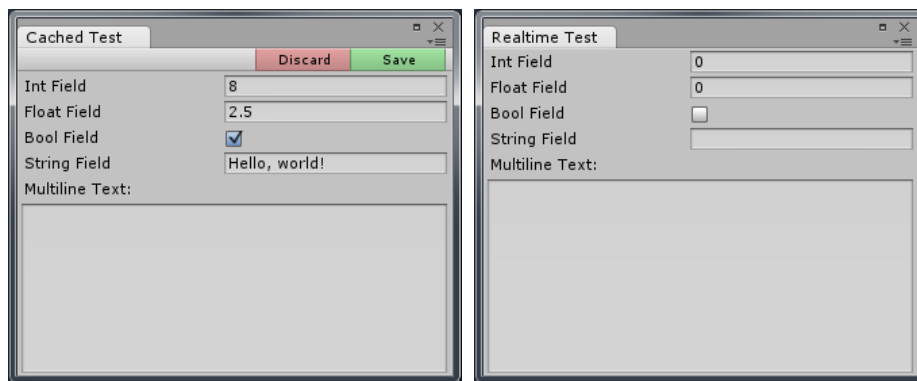
Access <https://github.com/rickyah/ini-parser> for further information.

V. USING THE DEMO

To use the demo just access the 'Window/ProjectPrefs' menu and open Cached Test window or Runtime Test window:



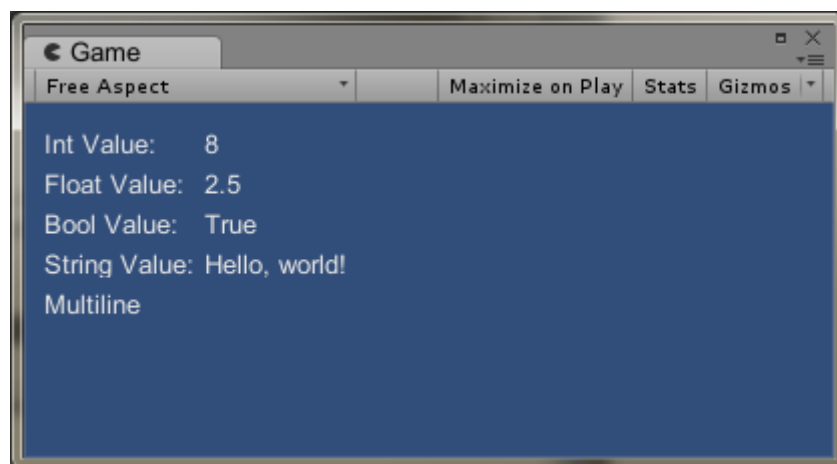
Tweak around with the values of any of these windows to see the interaction between them.



Observe that the first time you hit Save on Cached Test or change a value on Realtime Test an asset named 'Resources/prefs.ini' will be created if needed.

Feel free to look at the windows' source code. It's located at 'Bottle/Demo/Editor'.

Now open the scene located at 'Bottle/Demo/Scene' and hit play. You should see the following result:



Feel free to look at the GUI source code located at 'Bottle/Demo/GUITest'.